

## CODING DEVICE, CODING METHOD, DECODING DEVICE AND DECODING METHOD

Publication number: JP2001337699

Publication date: 2001-12-07

Inventor: KOYATA TOSHIHIRO

Applicant: SONY CORP

Classification:

- international: G10L11/00; G10L19/00; G10L19/02; H03M7/30; G10L11/00;  
G10L19/00; H03M7/30; (IPC1-7): G10L19/02; G10L11/00;  
G10L19/00; H03M7/30

- european:

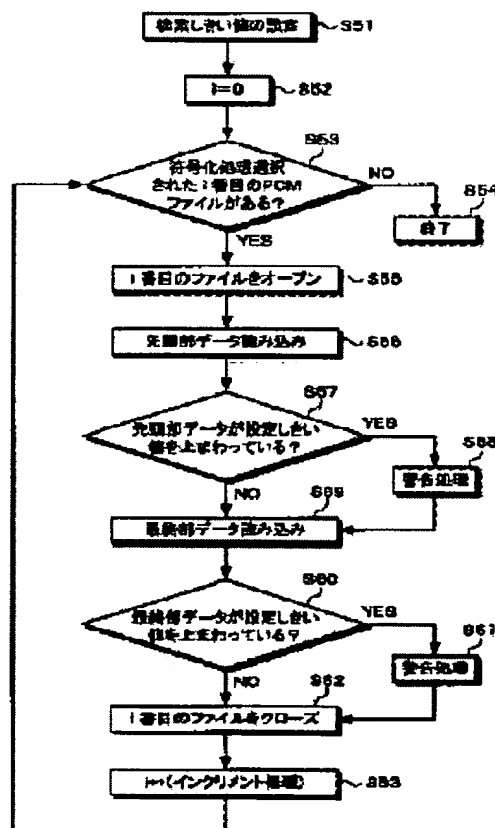
Application number: JP20000159931 20000530

Priority number(s): JP20000159931 20000530

Report a data error here

## Abstract of JP2001337699

**PROBLEM TO BE SOLVED:** To judge easily whether or not continuity among several files should be considered when the several files are coded or decoded. **SOLUTION:** A retrieval condition is set in a step S51 and a variable is initialized in a step S52. When it is determined that there is a file to be retrieved in a step S53, the file is opened in a step S55 and the front part data is read in a step S56. In a step S57 the condition of set threshold value is compared with the data read un. If the data read in is above the set threshold value, warning process is done in a step S58. In a step S59, S60 and S61 final part data is processed. In a step S62 the file is closed and in a step S63 i is incremented, returning to the step S53. Only the file that is given a warning is listened to and thus it is easy to judge whether or not the continuity should be considered.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号  
特開2001-337699  
(P2001-337699A)

(43)公開日 平成13年12月7日 (2001. 12. 7)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 1 0 L 19/02		H 0 3 M 7/30	A 5 D 0 4 5
11/00		G 1 0 L 7/04	G 5 J 0 6 4
19/00		9/16	
H 0 3 M 7/30		9/18	M

審査請求 未請求 請求項の数5 O L (全 28 頁)

(21)出願番号 特願2000-159931(P2000-159931)

(22)出願日 平成12年5月30日(2000. 5. 30)

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 小谷田 智弘

東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

(74)代理人 100082762

弁理士 杉浦 正知

Fターム(参考) 5D045 DA20

5J064 AA00 AA05 BA16 BB07 BB12

BC00 BC08 BC09 BC11 BC16

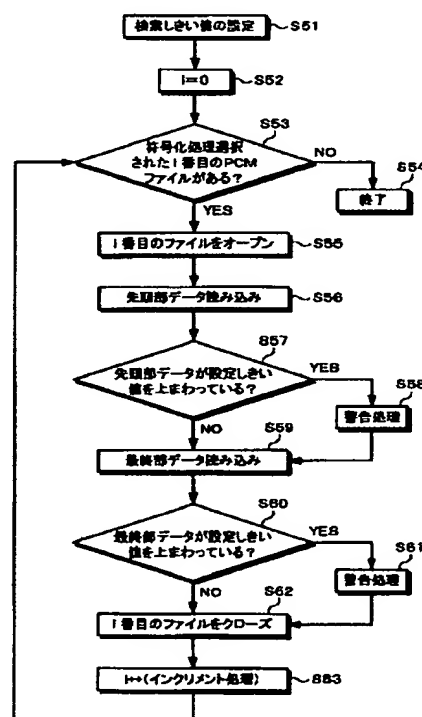
BC18 BC26 BC29 BD04

(54)【発明の名称】 符号化装置および符号化方法並びに復号化装置および復号化方法

(57)【要約】

【課題】 複数のファイルを符号化または復号化する時に、複数のファイル間の連続性を考慮すべきか否かを容易に判断できるようにする。

【解決手段】 ステップS51で検索条件が設定され、ステップS52で変数が初期化される。ステップS53で検索すべきファイルがあると決定されると、ステップS55でファイルがオープンされ、ステップS56で先頭部データが読み込まれる。ステップS57では、設定されたしきい値の条件と読み込んだデータが比較される。ステップS57で、読み込んだデータがしきい値を上回れば、ステップS58の警告処理を行う。ステップS59、S60、S61では、最終部データについての処理がなされる。ステップS62においてファイルをクローズし、ステップS63においてiをインクリメントし、ステップS53に戻る。警告のあったファイルについてのみ試験を行い、連続性を考慮すべきか否かを容易に判断することができる。



## 【特許請求の範囲】

【請求項 1】 複数のデジタルオーディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化装置であって、

上記複数のデジタルオーディオファイルの中から圧縮処理を施すデジタルオーディオファイルを選択する第 1 の選択手段と、

上記第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの前方に位置するデジタルオーディオファイルの終端部近傍のブロックと、上記第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの後方に位置するデジタルオーディオファイルの始端部近傍のブロックと、上記 2 つのデジタルオーディオファイルに跨がっているブロックとに基づいて符号化処理を施す第 1 の符号化手段と、

上記第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの前方に位置するデジタルオーディオファイルの終端部近傍のブロックと、上記 2 つのデジタルオーディオファイルに跨がっているブロックとに基づいて符号化処理を施す第 2 の符号化手段と、

上記複数のファイルの始点、終点のデータを分析する分析手段と、

上記分析手段の分析結果を参照して、上記第 1 の符号化手段における符号化処理と上記第 2 の符号化手段における符号化処理との一方を選択する第 2 の選択手段とを備えることを特徴とする符号化装置。

【請求項 2】 複数のデジタルオーディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化装置であって、

入力デジタル信号を複数の周波数帯域成分に分割する帯域分割手段と、

信号を直交変換して時間と周波数に関する複数の 2 次元ブロック内の符号化、および／または分析のための信号成分を得る直交変換手段と、

上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分を基に正規化を行なって正規化データを得る正規化データ算出手段と、

上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分の特徴を表す量子化係数を求める量子化係数算出手段と、

該量子化係数を基にビット配分量を決定するビット配算出手段と、

上記時間と周波数に関する 2 次元ブロック毎に上記正規化データとビット配分量によりブロック内の信号成分を量子化して情報圧縮する圧縮符号化手段と、

上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを得る情報圧縮パラメータ決定手段と、

処理対象の複数のファイルの始点、終点のデータを分析す

る分析手段と、

上記複数のファイルの処理時に、上記分析手段の分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択する処理選択手段とを備える符号化装置。

【請求項 3】 複数のデジタルオーディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化方法であって、

10 入力デジタル信号を複数の周波数帯域成分に分解して、時間と周波数に関する複数の 2 次元ブロック内の信号成分を得、上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分を基に正規化を行って正規化データを得、上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分の特徴を表す量子化係数を求め、該量子化係数を基にビット配分量を決定し、上記時間と周波数に関する 2 次元ブロック毎に上記正規化データとビット配分量によりブロック内信号成分を量子化した量子化データと、上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを、情報圧縮データとする符号化を行い、処理対象の複数のファイルの始点、終点のデータを分析し、上記複数のファイルの処理時に、分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択できるようにした符号化方法。

【請求項 4】 符号化された複数のデジタルオーディオファイルに対して復号化を施す復号化装置であって、入力デジタル信号を複数の周波数帯域成分に分割する帯域分割手段と、

30 信号を直交変換して時間と周波数に関する複数の 2 次元ブロック内の符号化、および／または分析のための信号成分を得る直交変換手段と、

上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分を基に正規化を行なって正規化データを得る正規化データ算出手段と、

上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分の特徴を表す量子化係数を求める量子化係数算出手段と、

40 該量子化係数を基にビット配分量を決定するビット配算出手段と、

上記時間と周波数に関する 2 次元ブロック毎に上記正規化データとビット配分量によりブロック内の信号成分を量子化して情報圧縮する圧縮符号化手段と、

上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを得る情報圧縮パラメータ決定手段と、

上記情報圧縮された時間と周波数に関する 2 次元ブロック内の信号成分を、上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを用いて復号する復号手段と、

50 処理対象の複数のファイルの始点、終点のデータを分析す

る分析手段と、

上記複数ファイルの処理時に、上記分析手段の分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択する処理選択手段とを備える符号化装置。

【請求項 5】 符号化された複数のデジタルオーディオファイルに対して復号化を施す復号化方法であって、入力デジタル信号を複数の周波数帯域成分に分解して、時間と周波数に関する複数の 2 次元ブロック内の信号成分を得、上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分を基に正規化を行って正規化データを得、上記時間と周波数に関する 2 次元ブロック毎に 2 次元ブロック内の信号成分の特徴を表す量子化係数を求め、該量子化係数を基にビット配分量を決定し、上記時間と周波数に関する 2 次元ブロック毎に上記正規化データとビット配分量によりブロック内信号成分を量子化した量子化データと、上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを、情報圧縮データとする符号化を行い、符号化された情報圧縮データの量子化データを、上記時間と周波数に関する 2 次元ブロック毎の情報圧縮パラメータを用いて復号化を行い、処理対象の複数ファイルの始点、終点のデータを分析し、上記複数ファイルの処理時に、分析結果を参照して、異なるファイル間の連続性を考慮した復号化と、考慮しない復号化を選択できるようにした復号化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、オーディオデータ等のデジタル信号に係る符号化装置および符号化方法並びに復号化装置および復号化方法に関する。

【0002】

【従来の技術】 オーディオ信号の高効率符号化に係る従来技術として、例えば、時間領域のオーディオ信号を単位時間毎にブロック化し、ブロック毎の時間軸上の信号を周波数軸上の信号に変換（直交変換）して複数の周波数帯域に分割し、各帯域毎に符号化するブロック化周波数帯域分割方式の一つである変換符号化方法が知られている。また、時間領域のオーディオ信号を単位時間毎にブロック化せずに、複数の周波数帯域に分割して符号化する非ブロック化周波数帯域分割方法の一つである帯域分割符号化（サブ・バンド・コーディング（SBC: Sub Band Coding））方法が知られている。

【0003】 さらに、上述の帯域分割符号化と変換符号化とを組み合わせる高能率符号化方法も知られている。この方法では、例えば、帯域分割符号化方式によって分割した各帯域毎の信号を、変換符号化方式によって周波数領域の信号に直交変換し、直交変換された各帯域毎に符号化が施される。

【0004】 ここで、上述した帯域分割符号化方式に使用される帯域分割用フィルタとしては、例えば QMF (Q

uadrature Mirror filter) 等のフィルタがある。QMF については、例えば、R.E.Crochiere Digital coding of speech in subbands Bell Syst. Tech. J. Vol.55, No.8(1976)に述べられている。また、ICASSP 83, BOSTON Polyphase Quadrature filters-A new subband coding technique Joseph H. Rothweiler には、ポリフェーズ クワドラチャ フィルタ (Polyphase Quadrature filter) などの等バンド幅のフィルタ分割手法および装置が述べられている。

10 【0005】 また、直交変換としては、例えば、入力オーディオ信号を所定単位時間（フレーム）でブロック化し、該ブロック毎に高速フーリエ変換（FFT）やコサイン変換（DCT）、モディファイド DCT 変換（MDCT）等を行うことで時間軸を周波数軸に変換するような方法が知られている。MDCT については、例えば、ICASSP 1987 Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation J.P.Princen A.B.Bradley Univ. of Surrey Royal Melbourne Inst. of Tech. に述べられている。

20 【0006】 一方、周波数帯域分割された各周波数成分を量子化する際に、人間の聴覚特性を考慮した周波数分割幅を用いる符号化方法が知られている。すなわち、臨界帯域（クリティカルバンド）と呼ばれる、帯域幅が高域程広くなるような帯域幅が広く用いられている。このような臨界帯域を用いてオーディオ信号を複数バンド（例えば 25 バンド）の帯域に分割することがある。このような帯域分割方法によれば、各帯域毎のデータを符号化する際に、各帯域毎に所定のビット配分、或いは各帯域毎に適応的なビット配分による符号化が行われる。

30 例えば、MDCT 処理によって生成される MDCT 係数データを上述したようなビット配分によって符号化する場合には、各ブロック毎に対応して生成される各帯域毎の MDCT 係数データに対して適応的なビット数が配分され、そのようなビット数配分の下で符号化が行われる。

【0007】 このようなビット配分方法およびそれを実現する装置についての公知文献として、例えば以下のようなものが挙げられる。まず、例えば IEEE Transactions of Acoustics, Speech, and Signal Processing, vol. AS SP-25, No. 4, August (1977) には、各帯域毎の信号の大きさに基づいてビット配分を行う方法が記載されている。また、例えば ICASSP 1980 The critical band coder--digital encoding of the perceptual requirements of the auditory system M.A. Kransner MIT には、聴覚マスキングを利用することによって各帯域毎に必要な信号対雑音比を得て固定的なビット配分を行う方法が記載されている。

【0008】 また、各帯域毎の符号化に際しては、各帯域毎に正規化を行って量子化を行うことにより、より効率的な符号化を実現するいわゆるブロックフローティン

グ処理が行われている。例えば、MDC T 処理によって生成される MDC T 係数データを符号化するには、各帯域毎に上述の MDC T 係数の絶対値の最大値等に対応した正規化を行った上で量子化を行うことにより、より効率的な符号化が行われる。正規化処理は例えば以下のように行われる。すなわち、予め番号付けされた複数種類の値を用意し、それら複数種類の値の内で各ブロックについての正規化に係るものを所定の演算処理によって決定し、決定した値に付されている番号を正規化情報として使用する。複数種類の値に対応する番号付けは、例えば、番号の 1 の増減に、オーディオレベルの 2 d B の増減が対応する等の一定の関係の下で行われる。

【0009】上述したような方法で生成される高能率符号化データは、次のようにして復号化される。まず、各帯域毎のビット配分情報、正規化情報等を参照して、符号化データに基づいて MDC T 係数データを生成する処理がなされる。この MDC T 係数データに基づいていわゆる逆直交変換 (IMDC T) が行われることにより、時間領域のデータが生成される。高能率符号化の過程で帯域分割用フィルタによる帯域分割が行なわれていた場合は、帯域合成フィルタを用いて時間領域のデータを合成する処理がさらになされる。

【0010】上述した符号化に用いられている直交変換の MDC T 処理、並びに復号化に用いられている、逆直交変換の IMDC T 処理では、処理を行うフレーム間の不連続性を防止するために、いわゆるオーバーラップ処理が利用されている。ある楽曲を符号化し、また、復号化する時には、当該楽曲の始点および終点については、このオーバーラップおよび変換サイズを考慮した適合処理が行われる。

【0011】上述した方法での高能率符号化は、基本的には楽曲単位で行われるが、大量の楽曲を高能率符号化処理するような場合、各楽曲の処理の終了毎に、ユーザが次の楽曲の処理の開始を促すのは非効率的であるため、通常、あらかじめ所望の楽曲を選択して、自動的に選択された楽曲が高能率符号化されるような処理が行われる。より具体的には、電子音楽配信の配信用サーバでは、ハードディスクに大量の PCM ファイルを格納し、コンピュータソフトウェア処理によって高速に高能率符号化の処理がなされる。

【0012】配信用サーバのように、大量の楽曲を自動的に高能率符号化処理する場合、楽曲単位で高能率符号化が行われるので、各楽曲に対して、始点および終点における、直交変換におけるオーバーラップおよび変換サイズを考慮した適合処理を行うことになる。楽曲によっては、他の楽曲との相関関係がある場合、例えば当該楽曲の始点が他の楽曲の終点との連続性を保つような場合がある。具体例としては、ライブ版、リミックス、ダンス系等の音楽では、楽曲同士が無音期間を介することなくつながっていることがある。このような場合でも、上

述したような始点および終点における適合処理を楽曲毎に独立して行くと、高能率符号化処理後のデータは、楽曲間の連続性を失ってしまう問題がある。復号化においても同様の問題が発生する。楽曲間に連続性があるものを処理する場合には、始点および終点における適合処理を行わずに、楽曲間データを連続的に処理することが望ましい。

#### 【0013】

【発明が解決しようとする課題】ところで、上述したような連続的な符号化を行うか否かの判断は、基本的には、実際に処理を行いたい各楽曲の先頭部分と最終部分について試聴確認することによって行われることになる。しかしながら、大量の楽曲を処理しなければならないような場合には、それらの全ての楽曲についての試聴確認を行う必要が生じ、多大な時間を費やすこととなる。また、長時間の確認作業に伴い、判断能力が低下し、判断の正確性が低下する可能性もある。

【0014】したがって、この発明の目的は、処理を行うべく選択された楽曲の、先頭部分と最終部分のデータを自動的に分析し、符号化または復号化に伴う連続性を考慮するか否かの判断を、容易、且つ正確に行い、作業効率を向上させることができる符号化装置および符号化方法並びに復号化装置および復号化方法を提供することにある。

#### 【0015】

【課題を解決するための手段】請求項 1 の発明は、複数のデジタルオーディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化装置であって、複数のデジタルオーディオファイルの中から圧縮処理を施すデジタルオーディオファイルを選択する第 1 の選択手段と、第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの前方に位置するデジタルオーディオファイルの終端部近傍のブロックと、第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの後方に位置するデジタルオーディオファイルの始端部近傍のブロックと、2 つのデジタルオーディオファイルに跨がっているブロックとに基づいて符号化処理を施す第 1 の符号化手段と、第 1 の選択手段にて選択された隣接するデジタルオーディオファイルの前方に位置するデジタルオーディオファイルの終端部近傍のブロックと、2 つのデジタルオーディオファイルに跨がっているブロックとに基づいて符号化処理を施す第 2 の符号化手段と、複数ファイルの始点、終点のデータを分析する分析手段と、分析手段の分析結果を参照して、第 1 の符号化手段における符号化処理と第 2 の符号化手段における符号化処理との一方を選択する第 2 の選択手段とを備えることを特徴とする符号化装置である。

【0016】請求項 2 の発明は、複数のデジタルオー

ディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化装置であって、入力デジタル信号を複数の周波数帯域成分に分割する帯域分割手段と、信号を直交変換して時間と周波数に関する複数の2次元ブロック内の符号化、および／または分析のための信号成分を得る直交変換手段と、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分を基に正規化を行なって正規化データを得る正規化データ算出手段と、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分の特徴を表す量子化係数を求める量子化係数算出手段と、該量子化係数を基にビット配分量を決定するビット配分算出手段と、時間と周波数に関する2次元ブロック毎に正規化データとビット配分量によりブロック内の信号成分を量子化して情報圧縮する圧縮符号化手段と、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを得る情報圧縮パラメータ決定手段と、処理対象の複数ファイルの始点、終点のデータを分析する分析手段と、複数ファイルの処理時に、分析手段の分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択する処理選択手段とを備える符号化装置である。

【0017】請求項3の発明は、複数のデジタルオーディオファイルに対して所定長毎にブロック化を施し、ブロック処理されたデジタルオーディオファイルに対して圧縮処理を施す符号化方法であって、入力デジタル信号を複数の周波数帯域成分に分解して、時間と周波数に関する複数の2次元ブロック内の信号成分を得、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分を基に正規化を行って正規化データを得、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分の特徴を表す量子化係数を求め、該量子化係数を基にビット配分量を決定し、時間と周波数に関する2次元ブロック毎に正規化データとビット配分量によりブロック内信号成分を量子化した量子化データと、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを、情報圧縮データとする符号化を行い、処理対象の複数ファイルの始点、終点のデータを分析し、複数ファイルの処理時に、分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択できるようにした符号化方法である。

【0018】請求項4の発明は、符号化された複数のデジタルオーディオファイルに対して復号化を施す復号化装置であって、入力デジタル信号を複数の周波数帯域成分に分割する帯域分割手段と、信号を直交変換して時間と周波数に関する複数の2次元ブロック内の符号化、および／または分析のための信号成分を得る直交変換手段と、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分を基に正規化を行なって正規化データを得る正規化データ算出手段と、時間と周波数

に関する2次元ブロック毎に2次元ブロック内の信号成分の特徴を表す量子化係数を求める量子化係数算出手段と、該量子化係数を基にビット配分量を決定するビット配分算出手段と、時間と周波数に関する2次元ブロック毎に正規化データとビット配分量によりブロック内の信号成分を量子化して情報圧縮する圧縮符号化手段と、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを得る情報圧縮パラメータ決定手段と、情報圧縮された時間と周波数に関する2次元ブロック内の信号成分を、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを用いて復号する復号手段と、処理対象の複数ファイルの始点、終点のデータを分析する分析手段と、複数ファイルの処理時に、分析手段の分析結果を参照して、異なるファイル間の連続性を考慮した符号化と、考慮しない符号化を選択する処理選択手段とを備える符号化装置である。

【0019】請求項5の発明は、符号化された複数のデジタルオーディオファイルに対して復号化を施す復号化方法であって、入力デジタル信号を複数の周波数帯域成分に分解して、時間と周波数に関する複数の2次元ブロック内の信号成分を得、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分を基に正規化を行って正規化データを得、時間と周波数に関する2次元ブロック毎に2次元ブロック内の信号成分の特徴を表す量子化係数を求め、該量子化係数を基にビット配分量を決定し、時間と周波数に関する2次元ブロック毎に正規化データとビット配分量によりブロック内信号成分を量子化した量子化データと、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを、情報圧縮データとする符号化を行い、符号化された情報圧縮データの量子化データを、時間と周波数に関する2次元ブロック毎の情報圧縮パラメータを用いて復号化を行い、処理対象の複数ファイルの始点、終点のデータを分析し、複数ファイルの処理時に、分析結果を参照して、異なるファイル間の連続性を考慮した復号化と、考慮しない復号化を選択できるようにした復号化方法である。

【0020】以上のような発明によれば、処理対象の複数ファイルの始点、終点のデータを分析することによって、複数ファイルの連続性に関する分析を行うことができる。分析結果を参照して、試聴を行うファイルを限定することが可能となる。それによって、処理の効率を向上できる。

#### 【0021】

【発明の実施の形態】この発明の一実施形態について、以下、図面を参照して説明する。一実施形態では、オーディオPCM信号等の入力デジタル信号を、帯域分割符号化(SBC)、適応変換符号化(ATC)および適応ビット割当の技術を用いて高能率符号化する。この高能率符号化技術について、図1を参照して説明する。

【0022】図1に示す高能率符号化装置では、入力デ

デジタル信号を複数の周波数帯域に分割すると共に、各周波数帯域毎に直交変換を行って、得られた周波数軸のスペクトルデータを、低域では、後述する人間の聴覚特性を考慮したいわゆる臨界帯域幅（クリティカルバンド）毎に、中高域では、ブロックフローティング効率を考慮して臨界帯域幅を細分化した帯域毎に、適応的にビット割当して符号化している。通常このブロックが量子化雑音発生ブロックとなる。さらに、一実施形態においては、直交変換の前に入力信号に応じて適応的にブロックサイズ（ブロック長）を変化させている。

【0023】例えばサンプリング周波数が44.1kHzの場合、入力端子100を介して0~22kHzのオーディオPCM信号がQMFフィルタ等の帯域分割フィルタ101に供給される。帯域分割フィルタ101は、供給される信号を0~11kHz帯域と11kHz~22kHz帯域とに分割する。11~22kHz帯域の信号はMDCT(Modified Discrete Cosine Transform)回路103およびブロック決定回路109、110、111に供給される。

【0024】また、0kHz~11kHz帯域の信号は帯域分割フィルタ102に供給される。帯域分割フィルタ102は、供給される信号を5.5kHz~11kHz帯域と0~5.5kHz帯域とに分割する。5.5~11kHz帯域の信号はMDCT回路104およびブロック決定回路109、110、111に供給される。また、0~5.5kHz帯域の信号は、MDCT回路105およびブロック決定回路109、110、111に供給される。帯域分割フィルタ101、102は、例えばQMFフィルタ等を用いて構成することができる。ブロック決定回路109は、供給される信号に基づいてブロックサイズを決定し、決定したブロックサイズを示す情報をMDCT回路103および出力端子113に供給する。

【0025】ブロック決定回路110は、供給される信号に基づいてブロックサイズを決定し、決定したブロックサイズを示す情報をMDCT回路104および出力端子115に供給する。ブロック決定回路111は、供給される信号に基づいてブロックサイズを決定し、決定したブロックサイズを示す情報をMDCT回路105および出力端子117に供給する。ブロックサイズブロック決定回路110、111、112は、供給される信号の時間特性、周波数分布に応じて適応的にブロックサイズ（ブロック長）を設定する。

【0026】MDCT回路103、104、105は、供給される信号に基づいてMDCT処理を行い、MDCT係数データまたは周波数軸上のスペクトルデータを生成する。MDCT回路103が生成する高域のMDCT係数データまたは周波数軸上のスペクトルデータは、ブロックフローティングの有効性を考慮して臨界帯域幅を細分化する処理を施された後に適応ビット割当符号化回

路106およびビット割当算出回路118に供給される。MDCT回路104が生成する中域のMDCT係数データまたは周波数軸上のスペクトルデータは、ブロックフローティングの有効性を考慮して臨界帯域幅を細分化する処理を施された後に適応ビット割当符号化回路107およびビット割当算出回路118に供給される。

【0027】MDCT回路105が生成する低域のMDCT係数データまたは周波数軸上のスペクトルデータは、臨界帯域（クリティカルバンド）毎にまとめる処理を施された後に適応ビット割当符号化回路108およびビット割当算出回路118に供給される。ここで、臨界帯域とは、人間の聴覚特性を考慮して分割された周波数帯域であり、ある純音の周波数近傍の同じ強さの狭帯域バンドノイズによって当該純音がマスクされる時に、当該狭帯域バンドノイズの帯域のことである。臨界帯域は、高域ほど帯域幅が広くなるという性質がある。0~22kHzの全周波数帯域は、例えば25のクリティカルバンドに分割されている。

【0028】ビット割当算出回路118は、供給されるMDCT係数データまたは周波数軸上のスペクトルデータ、およびブロックサイズ情報に基づいて、後述するようなマスキング効果等を考慮して上述の臨界帯域およびブロックフローティングを考慮した各分割帯域毎のマスキング量、エネルギーおよび或いはピーク値等を計算し、計算結果に基づいて各帯域毎にブロックフローティングの状態を示すスケールファクタ、および割当てビット数を計算する。計算された割当てビット数は、適応ビット割当符号化回路106、107、108に供給される。以下の説明において、ビット割当の単位とされる各分割帯域を単位ブロックと表記する。

【0029】適応ビット割当符号化回路106は、ブロック決定回路109から供給されるブロックサイズ情報、ビット割当算出回路118から供給される割当てビット数および正規化情報としてのスケールファクタ情報に応じて、MDCT回路103から供給されるスペクトルデータまたはMDCT係数データを再量子化（正規化して量子化）する処理を行う。かかる処理の結果として、高能率符号化データが生成される。この高能率符号化データは演算器120に供給される。適応ビット割当符号化回路107は、ブロック決定回路110から供給されるブロックサイズ情報、ビット割当算出回路118から供給される割当てビット数およびスケールファクタ情報に応じて、MDCT回路104から供給されるスペクトルデータまたはMDCT係数データを再量子化する処理を行う。かかる処理の結果として、高能率符号化データが生成される。この高能率符号化データが演算器121に供給される。

【0030】適応ビット割当符号化回路108は、ブロック決定回路110から供給されるブロックサイズ情報、ビット割当て算出回路118から供給される割当

ビット数およびスケールファクタ情報に応じて、MDC T回路105から供給されるスペクトルデータまたはMDC T係数データを再量子化する。かかる処理の結果として、高能率符号化データが生成される。この高能率符号化データは演算器122に供給される。正規化情報変更回路119、および演算器120、121、122については後述する。

【0031】図2に、MDC T回路103、104、105に供給される、各帯域毎のデータの例を示す。ブロック決定回路109、110、111の動作により、帯域分割フィルタ101、102から出力される計3個のデータについて、各帯域毎について独立に直交変換ブロックサイズを設定することができると共に、信号の時間特性、周波数分布等により時間分解能を切り換えることが可能とされている。すなわち、信号が時間的に準定常的である場合には、図2Aに示すような、直交変換ブロックサイズを例えば11.6msと大きくするLong Modeが用いられる。

【0032】一方、信号が非定常的である場合には、直交変換ブロックサイズをLong Mode時に比べて2分割または4分割とするモードが用いられる。より具体的には、全てを4分割して例えば2.9msとするShort Mode（図2B参照）、或いは、一部を2分割して例えば5.8msとし、他の一部を4分割して例えば2.9msとするMiddle Mode-a（図2C参照）または、Middle Mode-b（図2D参照）が用いられる。このように時間分解能を様々に設定することにより、実際の複雑な入力信号に適応できるようになる。

【0033】回路規模等に係る制約が小さい場合には、直交変換ブロックサイズの分割をさらに複雑なものとするにより、実際の入力信号をより適切に処理できることは明白である。上述したようなブロックサイズは、ブロック決定回路109、110、111によって決定され、決定されたブロックサイズの情報はMDC T回路103、104、105およびビット割り当て算出回路118に供給されると共に、出力端子113、115、117を介して出力される。

【0034】次に、図3を参照して、ビット割り当て算出回路118について詳細に説明する。入力端子301を介して、MDC T回路103、104、105からの周波数軸上のスペクトルデータまたはMDC T係数、およびブロック決定回路109、110、111からのブロックサイズ情報がエネルギー算出回路302に供給される。エネルギー算出回路302は、例えば当該単位ブロック内での各振幅値の総和を計算する等の方法で単位ブロック毎のエネルギーを計算する。なお、エネルギー算出回路302の代わりに振幅値のピーク値、平均値等を計算する構成を設け、振幅値のピーク値、平均値等の計算値に基づいてビット割り当て処理を行うようにしても良

い。

【0035】エネルギー算出回路302の出力の一例を図4に示す。図4では、各バンド毎の総和値のスペクトルSBを、先端に丸を付した縦方向の線分によって示す。ここで、横軸が周波数、縦軸が信号強度をそれぞれ示す。なお、図示が煩雑となるのを避けるため、図4では、単位ブロックによる分割数を12ブロック（B1～B12）とし、B12のスペクトルのみに符号「SB」を付した。

【0036】また、エネルギー算出回路302は、単位ブロックのブロックフローティングの状態を示す正規化情報であるスケールファクタ値を決定する処理を行う。具体的には、例えばあらかじめスケールファクタ値の候補として幾つかの正の値を用意し、それらの内、単位ブロック内のスペクトルデータ又はMDC T係数の絶対値の最大値以上の値をとるものの中で最小のものを当該単位ブロックのスケールファクタ値として採用する。スケールファクタ値の候補は、実際の値と対応した形で、例えば数ビットを用いて番号付けを行ない、その番号を図示しないROM（Read Only Memory）等に記憶させておけば良い。この際に、スケールファクタ値の候補は、番号順に例えば2dBの間隔での値を持つように規定しておく。ある単位ブロックについて採用されたスケールファクタ値に付される番号がサブ情報として用いられ、当該単位ブロックについてのスケールファクタ情報とされる。

【0037】エネルギー算出回路302の出力、すなわち、スペクトルSBの各値は、畳込みフィルタ回路303に送られる。畳込みフィルタ回路303は、例えば、入力データを順次遅延させる複数の遅延素子と、これら遅延素子からの出力にフィルタ係数（重み付け関数）を乗算する複数の乗算器と、各乗算器出力の総和をとる総和加算器とから構成することができる。畳込みフィルタ回路303は、スペクトルSBのマスクングにおける影響を考慮するための、スペクトルSBに所定の重み付け関数を掛けて加算するような畳込み（コンボリューション）処理を施す。この畳込み処理により、図4中で点線で示す部分の総和が計算される。

【0038】図3に戻り、畳込みフィルタ回路303の出力は演算器304に供給される。演算器304には、さらに、許容関数（マスクングレベルを表現する関数）が（n-a i）関数発生回路305から供給される。演算器304は、許容関数に従って、畳込みフィルタ回路303によって畳み込まれた領域における、許容可能なノイズレベルに対応するレベル $\alpha$ を計算する。ここで、許容可能なノイズレベル（許容ノイズレベル）に対応するレベル $\alpha$ とは、後述するように、逆コンボリューション処理を行うことによって、クリティカルバンドの各バンド毎の許容ノイズレベルとなるようなレベルである。レベル $\alpha$ の算出値は、許容関数を増減させることによ



て制御される。

【0039】すなわち、許容ノイズレベルに対応するレベル $\alpha$ は、クリティカルバンドのバンドの低域から順に与えられる番号を $i$ とすると、次の式(1)で求めることができる。

$$\text{【0040】 } \alpha = S - (n - a_i) \quad (1)$$

【0041】式(1)において、 $n$ 、 $a$ は定数で $a > 0$ 、 $S$ は畳込み処理されたスペクトルの強度であり、式(1)中 $(n - a_i)$ が許容関数となる。一例として $n = 38$ 、 $a = 1$ とすることができる。

【0042】演算器304によって計算されるレベル $\alpha$ が割算器306に伝送される。割算器306は、レベル $\alpha$ を逆コンボリューションする処理を行い、その結果としてレベル $\alpha$ からマスキングスペクトルを生成する。このマスキングスペクトルが許容ノイズスペクトルとなる。なお、逆コンボリューション処理を行う場合、一般的には複雑な演算が行われる必要があるが、この発明の一実施形態では、簡略化した割算器306を用いて逆コンボリューションを行っている。マスキングスペクトルは、合成回路307に供給される。合成回路307には、さらに、後述するような最小可聴カーブRCを示すデータが最小可聴カーブ発生回路312から供給される。

【0043】合成回路307は、割算器306の出力であるマスキングスペクトルと最小可聴カーブRCのデータとを合成することにより、マスキングスペクトルを生成する。生成されるマスキングスペクトルが減算器308に供給される。減算器308には、さらに、エネルギー検出回路302の出力、すなわち帯域毎のスペクトルSBが遅延回路309によってタイミングを調整された上で供給される。減算器308は、マスキングスペクトルとスペクトルSBとに基づく減算処理を行う。

【0044】かかる処理の結果として、ブロック毎のスペクトルSBの、マスキングスペクトルのレベル以下の部分がマスキングされる。図5に、マスキングの一例を示す。スペクトルSBにおける、マスキングスペクトルのレベル(MSと表記する)以下の部分がマスキングされていることがわかる。なお、図示が煩雑となるのを避けるため、図5中ではB12においてのみ、スペクトルに符号「SB」を付すと共にマスキングスペクトルのレベルに符号「MS」を付した。

【0045】雑音絶対レベルが最小可聴カーブRC以下ならばその雑音は人間には聞こえないことになる。最小可聴カーブは、コーディングが同じであっても例えば再生時の再生ボリュームの違いによって異なる。但し、実際のデジタルシステムでは、例えば16ビットダイナミックレンジへの音楽データの入り方にはさほど違いがないので、例えば4kHz付近の最も耳に聞こえやすい周波数帯域の量子化雑音が聞こえないとすれば、他の周波数帯域ではこの最小可聴カーブのレベル以下の量子化雑

音は聞こえないと考えられる。

【0046】従って、例えばシステムの持つワードレングスの4kHz付近の雑音が聞こえないような使い方をする場合、最小可聴カーブRCとマスキングスペクトルMSとを合成することによって許容ノイズレベルを得るようにすれば、この場合の許容ノイズレベルは図6中の斜線で示す部分となる。なお、ここでは、最小可聴カーブの4kHzのレベルを例えば20ビット相当の最低レベルに合わせている。図6では、各ブロック内の水平方向の実線としてSB、各ブロック内の水平方向の点線としてMSをそれぞれ示した。但し、図示が煩雑となるのを避けるため、図6ではB12のスペクトルのみについて符号「SB」、「MS」を付した。また、図6では、信号スペクトルSSを一点鎖線で示した。

【0047】図3に戻り、減算器308の出力は許容雑音補正回路310に供給される。許容雑音補正回路310は、例えば等ラウドネスカーブのデータ等に基づいて、減算器308の出力における許容雑音レベルを補正する。すなわち、許容雑音補正回路310は、上述したマスキング、聴覚特性等の様々なパラメータに基づいて、各単位ブロックに対する割当ビットを算出する。許容雑音補正回路310の出力は、出力端子311を介して、ビット割当算出回路118の最終的な出力データとして出力される。ここで、等ラウドネスカーブとは、人間の聴覚特性に関する特性曲線であり、例えば1kHzの純音と同じ大きさに聞こえる各周波数での音の音圧を求めて曲線で結んだもので、ラウドネスの等感度曲線とも呼ばれる。

【0048】また、この等ラウドネスカーブは、図6に示した最小可聴カーブRCと同じ曲線を描く。この等ラウドネスカーブにおいては、例えば4kHz付近では1kHzのところより音圧が8~10dB下がっても1kHzと同じ大きさに聞こえ、逆に、50Hz付近では1kHzでの音圧よりも約15dB高くなくとも同じ大きさに聞こえない。このため、最小可聴カーブRCのレベルを越える雑音(許容ノイズレベル)が等ラウドネスカーブに沿った周波数特性を持つようにすれば、その雑音が人間に聞こえないようにすることができる。

【0049】等ラウドネスカーブを考慮して許容ノイズレベルを補正することは、人間の聴覚特性に適合していることがわかる。以上のように、ビット割当算出回路118では、メイン情報としての直交変換出力スペクトルをサブ情報によって処理したデータと、サブ情報としてのブロックフローティングの状態を示すスケールファクタおよび語調を示すワードレングスが得られる。これらの情報に基づいて、図1中の適応ビット符号化回路106、107、108が再量子化を行って、符号化フォーマットに従う高能率符号化データを生成する。

【0050】図1に戻り、正規化情報変更回路119について説明する。上述したように、エネルギー算出回路

302によって決定されるスケールファクタ情報を操作することにより、例えば2dB毎のレベル調整を行うことができる。正規化情報変更回路119は、スケールファクタ情報の変更に係る値を生成し、生成した値をそれぞれ、演算器120、121、122に供給する。演算器120は、121、122は、それぞれ、適応ビット割当符号化回路106、107、108から供給される符号化データ中のスケールファクタ情報に、正規化情報変更回路119から供給される値を加算する。但し、正規化情報変更回路119から出力される値が負の場合は、演算器120、121、122は減算器として作用するものとする。この際に加算結果については、フォーマットで定められたスケールファクタの数値の範囲内に収まるような制限を行う。

【0051】なお、スケールファクタ情報に加算すべき値として、正規化情報変更回路119が全単位ブロックに対して同一の値を出力する場合にはレベル調整処理が行われるが、正規化情報変更回路119が単位ブロック毎に異なる値を出力するようにすれば、例えばフィルタ処理等を実現できる。フィルタ処理等を行う場合には、正規化情報変更回路119は、スケールファクタ情報に加算すべき値と、その値が加算されるべきスケールファクタ情報をに係る単位ブロックの番号との組を出力する。以上のような正規化情報調整処理は、後述する復号化の場合に実現することも可能である。

【0052】次に、高能率符号化データの符号化フォーマットについて、図7を参照して説明する。左側に示した数値0、1、2、・・・、211はバイト数を表しており、この一例では212バイトを1フレームの単位としている。先頭の0バイト目の位置には、図1中のブロック決定回路109、110、111において決定された、各帯域のブロックサイズ情報を記録する。次の1バイト目の位置には、記録する単位ブロックの個数の情報を記録する。例えば高域側になる程、ビット割当算出回路118によってビット割当が0とされて記録が不必要となる場合が多いため、このような状況に対応するように単位ブロックの個数を設定することにより、聴感上の影響が大きい中低域に多くのビットを配分するようになっている。それと共に、かかる1バイト目の位置にはビット割当情報の2重書きを行なっている単位ブロックの個数、及びスケールファクタ情報の2重書きを行なっている単位ブロックの個数が記録される。

【0053】2重書きとは、エラー訂正用に、あるバイト位置に記録されたデータと同一のデータを他の場所に記録する方法である。2重書きされるデータの量を多くする程、エラーに対する強度が向上するが、2重書きされるデータの量を少なくする程、スペクトラムデータに使用できるデータ容量が多くなる。この符号化フォーマットの一例では、ビット割当情報、スケールファクタ情報のそれぞれについて独立に2重書きを行なう単位ブ

ックの個数を設定することにより、エラーに対する強度と、スペクトラムデータを記録するために使用されるビット数とを適切なものとするようにしている。なお、それぞれの情報について、規定されたビット内でのコードと単位ブロックとの個数の対応は、あらかじめフォーマットとして定めている。

【0054】1バイト目の位置の8ビットにおける記録内容の一例を図8に示す。ここでは、最初の3ビットを実際に記録される単位ブロックの個数の情報とし、後続の2ビットをビット割当情報の2重書きを行なっている単位ブロックの個数の情報とし、最後の3ビットをスケールファクタ情報の2重書きを行なっている単位ブロックの個数の情報とする。

【0055】図8において、2バイト目からの位置には、単位ブロックのビット割当情報が記録される。ビット割当情報の記録のために、単位ブロック1個当たり例えば4ビットが使用される。これにより、0番目の単位ブロックから順番に記録される単位ブロックの個数分のビット割当情報が記録されることになる。ビット割当情報のデータの後に、各単位ブロックのスケールファクタ情報が記録される。スケールファクタ情報の記録のために、単位ブロック1個当たり例えば6ビットが使用される。これにより、0番目の単位ブロックから順番に記録される単位ブロックの個数分のスケールファクタ情報が記録される。

【0056】スケールファクタ情報の後に、単位ブロック内のスペクトラムデータが記録される。スペクトラムデータは、0番目の単位ブロックより順番に、実際に記録させる単位ブロックの個数分記録される。各単位ブロック毎に何本のスペクトラムデータが存在するかは、あらかじめフォーマットで定められているので、上述したビット割当情報によりデータの対応をとることが可能となる。なお、ビット割当が0の単位ブロックについては記録を行わない。

【0057】このスペクトラム情報の後に、上述したスケールファクタ情報の2重書き、およびビット割当情報の2重書きを行なう。この2重書きの記録方法は、個数の対応を図8に示した2重書きの情報に対応させるだけで、その他の点については上述のスケールファクタ情報、およびビット割当情報の記録と同様である。最後のバイトすなわち211バイト目、およびその1バイト前の位置すなわち210バイト目には、それぞれ、0バイト目と1バイト目の情報が2重書きされる。これら2バイト分の2重書きはフォーマットとして定められており、スケールファクタ情報の2重書きやビット割当情報の2重書きのように、2重書き記録量の可変の設定はできない。

【0058】次に、高能率符号化データを復号化する復号化処理について説明する。復号化処理系の構成の一例を図9に示す。高能率符号化データは、入力端子707

を介して演算器 710 に供給される。また、符号化処理において使用されたブロックサイズ情報、すなわち図 1 中の出力端子 113、115、117 の出力信号と等価のデータが入力端子 708 に供給される。また、正規化情報変更回路 709 は、各単位ブロックのスケールファクタ情報に加算または減算すべき値を生成する。

【0059】演算器 710 は、さらに、正規化情報変更回路 709 から数値データを供給される。演算器 710 は、供給される高能率符号化データ中のスケールファクタ情報に対して、正規化情報変更回路 709 から供給される数値データを加算する。但し、正規化情報変更回路 709 から供給される数値データが負の数の場合は、演算器 710 は減算器として作用するものとする。演算器 710 の出力は、適応ビット割当復号化回路 706、および出力端子 711 に供給される。

【0060】適応ビット割当復号化回路 706 は、適応ビット割当情報を参照してビット割当てを解除する処理を、高域、中域、低域の各帯域について行う。高域、中域、低域のそれぞれに対する適応ビット割当て復号化回路 706 の出力は、逆直交変換回路 703、704、705 に供給される。逆直交変換回路 703、704、705 は、供給されるデータを逆直交変換処理する。これにより、周波数軸上の信号が時間軸上の信号に変換される。逆直交変換回路 703、704、705 の出力である、部分帯域の時間軸上信号は、帯域合成フィルタ 701、702 によって合成され、全帯域信号に復号化される。帯域合成フィルタ 701、702 としては、例えば IQMF (Inverse Quadrature Mirror filter) 等を使用することができる。

【0061】演算器 710 による加算または減算によってスケールファクタ情報を操作することにより、再生データについて例えば 2 dB 毎のレベル調整を行うことができる。例えば、正規化情報変更回路 709 から全て同じ数値を出力し、その数値を全単位ブロックのスケールファクタ情報に一律に加算または減算する処理により、全単位ブロックに対して 2 dB を単位とするレベル調整を行うことが可能とされる。

【0062】また、例えば、正規化情報変更回路 709 から単位ブロック毎に独立な数値を出力し、それらの数値を各単位ブロックのスケールファクタ情報に加算または減算する処理によって単位ブロック毎のレベル調整を行うことができ、その結果としてフィルタ機能を実現することができる。より具体的には、正規化情報変更回路 709 が単位ブロックの番号と、当該単位ブロックのスケールファクタ情報に加算または減算すべき値との組を出力させる等の方法で、単位ブロックと当該単位ブロックのスケールファクタ情報に加算または減算すべき値とが対応付けられるようにする。なお、演算器 710 による加算または減算の結果として生成されるスケールファクタ情報は、対応するスケールファクタ値が高能率符号

化データのフォーマットで定められた範囲に収まるように制限される。

【0063】演算器 710 によって単位ブロックのレベル調整が行われたスケールファクタ値については、適応ビット割当復号化回路 706 の復号化の行程に使用することにより、復号化信号のレベル調整を行うのみに利用することが可能であると共に、例えば符号化情報が記録された記録媒体よりスケールファクタ値を読み込み、調整が行われたスケールファクタ値を出力端子 711 に出力させ、記録媒体に記録されたスケールファクタ値を調整された値に変更することも可能である。記録媒体の情報の変更については、必要に応じて行えるものとする。これによって、非常に簡単なシステムで、記録媒体のレベル情報を変更することが可能となる。

【0064】上述の説明では、符号化回路、復号化回路の双方においてスケールファクタ情報の変更処理を行うものとした。これに対して、復号化回路のみにおいてスケールファクタ情報の変更処理を行うようにした場合にも、変更処理の結果として、レベル調整、フィルタ処理等の機能を充分に得ることができる。

【0065】次に、上述した高能率符号化における処理を行う時間単位について説明する。図 1 における入力端子 100 には、オーディオの PCM サンプルが供給されるが、入力後に行われる MDCT 回路 103、104、105 による MDCT 処理においては、いわゆる直交変換処理を行うためのサンプル数が規定され、それが一つの単位となり、繰り返し処理がなされる。

【0066】ここでは、入力端子 100 から入力された 1024 サンプルの PCM サンプルが 512 本の MDCT 係数、またはスペクトラムデータとして、MDCT 回路 103、104、105 より出力される。具体的には、入力端子 100 から入力された 1024 個の PCM サンプルが帯域分割フィルタ 101 によって、512 個の高域サンプルと 512 個の低域サンプルと 256 個の中域サンプルとなる。その後、帯域分割フィルタ 102 からの 256 個の低域サンプルは、MDCT 回路 105 によって、128 個の低域スペクトラムデータとなり、帯域分割フィルタ 102 からの 256 個の中域サンプルは、MDCT 回路 104 によって、128 個の中域スペクトラムデータとなり、帯域分割フィルタ 101 からの 512 個の高域サンプルは、MDCT 回路 103 によって、256 個の高域スペクトラムデータとなる。このように、合計 512 個のスペクトラムデータが 1024 個の PCM サンプルから作成される。この 1024 個の PCM サンプルが上述した高能率符号化の 1 回の処理を行う時間単位となり、図 7 に示した 212 バイトの高能率符号化データ、すなわち、1 フレームとなる。

【0067】上述したように、1 フレームは、例えば 1024 個の PCM サンプルからなるが、図 1 中の MDCT 回路 103、104、105 による MDCT 処理にお

いては、通常、順次処理されていく各フレームにおいてオーバーラップ部分が生じる。PCMサンプルとフレームの関係を図10を用いて説明する。図10に示すように、例えば、 $n$ 番目から $n+1023$ 番目までの1024個のPCMサンプルが $N$ 番目のフレームで処理される場合に、 $N+1$ 番目のフレームでは、 $n+512$ 番目から $n+1535$ 番目までの1024個のPCMサンプルが処理され、 $N+2$ 番目のフレームでは、 $n+1024$ 番目から $n+2047$ 番目までの1024個のPCMサンプルが処理される。このように、一つのフレームは、隣接するサウンドフレームと、512個のPCMサンプルのオーバーラップを持つ形となる。つまり、このような形で処理を行うと、高能率符号化情報の1フレームは、1024個のPCMサンプルを処理したものであるが、隣接フレームとのオーバーラップを考慮すると、512個のPCMサンプル相当ということになる。

【0068】図10は、PCMサンプルの途中でのフレームとの対応を示しているが、PCMサンプルの始点については、例えば始点より以前の段階に512個の0データのPCMサンプルを想定して、これらの512個の0データのPCMサンプルを、最初のフレーム以前の仮想的なフレームとオーバーラップして処理するものとする。また、最後のフレームでは、サンプル列終了時点以後に512個の0データのPCMサンプルを想定して、それら512個の0データのPCMサンプルを、最後のフレーム以後の仮想的なフレームとオーバーラップして処理するものとする。

【0069】次に、上述した符号化または復号化方法について、いわゆるパソコン上のソフトウェアとして処理する方法について説明する。パソコン上での処理としては、主にハードディスク上のPCMのデータファイルを高能率符号化することにより、ハードディスク上に高能率符号化データファイルを作成する、またはハードディスク上の高能率符号化データファイルを復号化処理することによりハードディスク上にPCMのデータファイルを作成することが考えられる。この時、通常一つの楽曲が一つのファイルに対応される。

【0070】具体例として、いわゆるパソコンにおける、GUI(Graphical User Interface)を利用したソフトウェアでの画面表示、操作方法、処理行程等について、図11を用いて説明する。図11は、符号化および復号化のソフトウェアのパソコン上での画面表示の一例を示すものである。このソフトウェアは、まずPCMデータと高能率符号化データのためのディレクトリを選択する。801は、PCMデータファイルのディレクトリパスの表示部であり、現在この例ではCドライブのPCMDATAという名のディレクトリが選択されていることが示されている。803は、表示部801にて示されたディレクトリ内のファイル構成を表示すると共に、ディレクトリ移動、ドライブ移動、ファイル選択等を行え

る表示操作部である。この例では、現在の表示部801で示されたディレクトリの下には更にtmpという名称のディレクトリが存在していることが分かる。

【0071】また、「・」の表示は、一つ上の階層のディレクトリを示しているものとする。また、tmp以下6つのファイルはPCMデータファイルを示している。また、その下の[-c-] [-d-]は、移動可能なドライブを示している。表示されているものが、ディレクトリか、ドライブか、PCMデータかの判断は、表示されている文字列や、文字列の横に付加されている、いわゆるアイコンにより、判断することが可能である。

【0072】ディレクトリとドライブの表示部は、その文字列位置にマウスポインタを対応させ、ダブルクリックすることで、現行ディレクトリ位置を、ダブルクリックした場所に移動させることが可能である。この例では、例えばtmpの場所でダブルクリックを行うと、表示部801の表示は、C:¥PCMDATA¥tmpとなり、表示操作部803では、tmpの下の子ファイルの状態、および移動可能ドライブが示されるようになる。このように、ドライブ名やディレクトリ名をダブルクリックを繰り返すことにより、PCMデータファイル用の所望のディレクトリ位置に移動することができる。

【0073】802は、高能率符号化データ用のディレクトリ位置を表示する表示部であり、図示の例では、CドライブのENCODEDATAという名のディレクトリが選択されていることが示されている。804は、表示部802にて示されたディレクトリ内のファイル構成を表示すると共に、ディレクトリ移動、ドライブ移動、ファイル選択等を行える表示操作部である。この例では、表示部802で示された現在のディレクトリの下には、ファイル、ディレクトリが共に存在していないことが示されている。表示操作部804における操作、および表示部802との対応については、表示操作部803、表示部801におけるものと同様であり、表示操作部804にて高能率符号化データ用のディレクトリを選択することができる。

【0074】805は、高能率符号化を実行するボタンであり、ここをクリックすることで、表示操作部803にて選択されたPCMデータファイルが順に高能率符号化され、表示部802で示されたディレクトリの下に高能率符号化ファイルが作成される。この実際の処理の流れについて図12を用いて説明する。

【0075】図12Aに示す状態では、図11における表示操作部803にて、data2.pcm、dataA.pcm、dataB.pcmの3つのPCMファイルが選択され、反転表示されている。ここで図11におけるボタン805をクリックすることにより、これらの3つのファイルがそれぞれ順に高能率符号化される。通常の高能率符号化処理の場合、処理を行うファイルの順序は特に問題とならない。

【0076】図12Bに示す状態では、高能率符号化処理実行中の表示画面を示すものであり、符号化処理行程の進行状況が、棒グラフのような形で認識できるようになっている。ここでは図示していないが、ボタンの形で処理を途中で中止するような手段を設けても良い。図12Cは、選択された全てのファイルの高能率符号化処理が終了した状態を示すものである。図11における操作表示部804には、処理により作成された3つの高能率符号化データファイル、data2enc.dat、dataAenc.dat、dataBenc.datが表示されている。処理後の、高能率符号化データファイルのファイル名については任意性があるが、ここでは処理を行うPCMファイル名の、いわゆる拡張子部分となる.pcmを取り除いた部分の名称にenc.datが自動的に付加されたファイル名を採用するようにしている。

【0077】次にボタン807について説明する。このボタン807は、複数のファイルの高能率符号化処理を、データ列として連続に扱うようにするものである。図12Bを参照して説明したように、ファイルを連続して処理する場合、一つつのファイルについて、図1による行程と、図10で示したデータ関係による処理を行うこととなる。このため、処理を行う全てのファイルについて、上述したように、始点での512個の0データのPCMサンプルの想定、および終点についての0データのPCMサンプルの想定を考慮した処理を行うこととなる。通常、楽曲がファイル毎に独立している場合はこの方法で問題とならないが、楽曲としては別であるがPCMデータとして連続となっているような場合、高能率符号化処理を行うことで、連続性が失われてしまうこととなる。

【0078】この例を、先に示した図12におけるdataA.pcm、dataB.pcmが連続したPCMデータである場合を想定し、図13A、図13B、および図13Cを用いて説明する。図13Aでは、分割点を境にして、dataA.pcmの終点のPCMデータとdataB.pcmの始点のPCMデータが連続しているものである様子を示している。

【0079】また、先に図10等を用いて説明した高能率符号化処理を行うフレーム割りの最終部分については、図13AにおけるNとN+1のような状態となったものとする。この時、dataA.pcmの最終部の処理を示したものが図13Bである。すなわち、N+1番目のフレームが最終フレームとなるが、図13Aにおける分割点以降のデータについては別ファイルのデータであるので、分割点以降のデータを使用せず端数分となった部分については0データを詰め込んで処理を行う。

【0080】これに対して、dataB.pcmの始点のデータについては、図13Cに示した形の処理を行う。すなわち、図13Aにおける分割点以前のデータに

については別ファイルのデータであるので、分割点以前のデータを使用せず、先頭フレームの1024個のPCMデータは、512個のゼロデータと512個のdataB.pcmの始点のデータから構成される。

【0081】この時、図13Bで示したdataA.pcmを処理するフレーム割りと、図13Cで示したdataB.pcmを処理するフレーム割りが異なったものとなる。また、それぞれが端数分としてゼロデータを挿入しているため、連続性も失われた状態となっている。すなわち、dataA.pcmとdataB.pcmを連続再生した場合は、連続した音となるが、dataAenc.datとdataBenc.datを復号化して連続再生した場合は、音切れのような形となってしまう。

【0082】これに対して、図11におけるボタン807をクリックして、データ列を連続した形で処理する場合の例を図14A、図14Bおよび図14Cを用いて説明する。図14Aに示すように、ファイルの分割点、およびdataA.pcmの処理フレーム割り等は、図13Aと同様の状態となっている。図14Bは、dataA.pcmの最終フレームの様子を示すものであるが、図13Bとは異なり、分割点より外側のデータに0データを埋めるのではなく、dataB.pcmのデータを採用している。

【0083】また、図14Cは、dataB.pcmの先頭のフレーム割りを示しているが、図13Cのように、ファイルの始点にフレームをあわせて0データを埋めるのではなく、dataA.pcmのフレーム割りと連続性を保つようなフレーム割り処理として、dataB.pcmの始点より外側のデータについては、dataA.pcmのデータを採用するようにしている。つまり図14Aでのフレーム割りで考えた場合の、N+2というのがdataB.pcmの先頭フレームということになる。このように処理することにより、高能率符号化処理データにおいても、二つのファイル間で連続性が保たれることとなり、dataAenc.datとdataBenc.datを復号化して連続再生した場合の音切れが起こらないこととなる。

【0084】上述した図13A、図13B、図13Cに示したように、符号化処理を行う場合の処理を図15のフローチャートに示し、図14A、図14B、図14Cに示したように、符号化処理を行う場合の処理を図16のフローチャートに示す。

【0085】図15の最初のステップS1では、1024ポイント分の読み込みバッファを用意する。次に、処理の対象のファイルの番号iを0に設定する(ステップS2)。ステップS3では、処理すべきi番目のファイルがあるかどうか決定される。ファイルがなければ、処理は、終了する(ステップS4)。

【0086】i番目のファイルがある場合に、ステップ

S5において、読み込みバッファの前半512ポイント分データとしてゼロデータを詰める処理を行う。次に、 $i$  番目の読み込みファイル（PCMファイル）をオープンし（ステップS6）、そして、 $i$  番目の書き込みファイル（符号化ファイル）をオープンする（ステップS7）。読み込んだ符号化からバッファの後半512ポイントにデータを読み込む（ステップS8）。

【0087】ステップS9では、読み込みデータ量が取得され、読み込み位置が更新される。ステップS10では、読み込みデータ量が512ポイントに満たないかどうか決定される。読み込みデータ量が512ポイントに満たない場合には、ステップS11において、読み込みバッファの512ポイントと、読み込みデータ量の差分量のデータとしてゼロデータが詰められる。

【0088】ステップS10で読み込みデータ量が512ポイントある場合、またはステップS11（ゼロデータの詰め込み）に続いて、ステップS12において、1フレーム分の符号化処理がなされる。ステップS13では、符号化データを書き込みファイルに書き込む。

【0089】ステップS10の決定の結果が肯定の場合（読み込みデータ量が512ポイントに満たない場合）では、ステップS14で、 $i$  番目の読み込みファイルをクローズし、ステップS15で $i$  番目の書き込みファイルをクローズし、ステップS16で $i$  のインクリメント処理がなされる。そして、処理がステップS3（ $i$  番目のファイルの有無の決定）に戻る。

【0090】ステップS10の決定の結果が否定の場合（読み込みデータ量が512ポイントある場合）では、ステップS13に続いてステップS17の処理がなされる。ステップS17では、読み込みバッファの後半512ポイント分のデータをその前半512ポイントにシフトする。そして、処理がステップS9（読み込みデータ量の取得、および読み込み位置の更新）に戻る。

【0091】このようにして、図13に示すように、楽曲がファイル毎に独立している場合に適用される処理がなされる。また、楽曲としては別であるが、PCMデータとして連続となっているような場合に適用される処理（図14）を図16のフローチャートを参照して説明する。

【0092】最初のステップS21で、1024ポイント分の読み込みバッファが用意される。ステップS22では、 $i$  が0に初期化される。ステップS23では、最初のファイル（ $i=0$ ）であるか否かが決定される。最初のファイルの場合には、ステップS24において、読み込みバッファの前半512ポイント分のデータとしてゼロデータが詰められる。そして、 $i$  番目の読み込み（PCM）ファイルをオープンし（ステップS25）、 $i$  番目の書き込み（符号化）ファイルをオープンする（ステップS26）。ステップS27では、読み込みファイルからバッファの後半の512ポイントにデータを

読み込む。

【0093】ステップS28では、読み込みデータ量が取得され、読み込み位置が更新される。ステップS29では、読み込みデータ量が512ポイントに満たないかどうか決定される。読み込みデータ量が512ポイントに満たない場合には、ステップS30において、処理すべき $i+1$  番目のファイルがあるかどうか決定される。

【0094】ステップS30において、処理すべき $i+1$  番目のファイルがないと決定されると、ステップS31では、読み込みバッファの512ポイントと、読み込みデータ量の差分量のデータとしてゼロデータが詰められる。

【0095】ステップS30において、処理すべき $i+1$  番目のファイルがあると決定されると、ステップS32において、 $i+1$  番目の読み込みファイルのオープンがなされる。そして、ステップS33では、読み込みバッファの512ポイントと、読み込みデータ量の差分量のデータが $i+1$  番目のファイルから読み込まれ、読み込み位置が更新される。

【0096】ステップS29で読み込みデータ量が512ポイントある場合、ステップS31（ゼロデータの詰め込み）、またはステップS33（ $i+1$  番目のファイルからのデータの読み込みと、読み込み位置の更新）に続いて、ステップS34において、1フレーム分の符号化処理がなされる。ステップS35では、符号化データを書き込みファイルに書き込む。

【0097】ステップS29の決定の結果が否定の場合（読み込みデータ量が512ポイントある場合）では、ステップS35に続いてステップS36の処理がなされる。ステップS36では、読み込みバッファの後半512ポイント分のデータをその前半512ポイントにシフトする。そして、処理がステップS27（読み込みファイルからバッファの後半512ポイントにデータを読み込む）に戻る。

【0098】ステップS29の決定の結果が肯定の場合（読み込みデータ量が512ポイントに満たない場合）では、ステップS37で、 $i$  番目の読み込みファイルをクローズし、ステップS38で $i$  番目の書き込みファイルをクローズする。そして、ステップS30の決定の結果が否定（すなわち、 $i+1$  番目のファイルがない）場合に、処理が終了する（ステップS40）。一方、ステップS30の決定の結果が肯定（すなわち、 $i+1$  番目のファイルがある）場合に、ステップS39で $i$  のインクリメント処理がなされ、ステップS36の処理がなされる。そして、処理がステップS23（最初のファイルか否かの決定）に戻る。

【0099】次に図11におけるボタン809について説明する。ボタン809は、ファイルのデータを解析し、上述した連続性を考慮した符号化処理を行うべきか

否かの判断を行うためのものである。連続性の考慮については、通常、考慮すべきか否かの判断は、実際に楽曲ファイルの最終部、先頭部を試聴して判断されている。しかしながら、大量の楽曲ファイルが存在する場合、その全てについて試聴を行うことは、時間を浪費し、効率的ではない。

【0100】一般的にある楽曲ファイルがその一つのファイルで完結して、他のファイルとの連続性がないような場合、ファイルの先頭部付近、あるいは最終部付近のデータが無音となり、ゼロかゼロに近い値となる傾向にある。一方、他のファイルとの連続性があるようなファイルの場合、ファイルの先頭部付近、あるいは最終部付近のデータはゼロ以外で、ある程度の大きさ、すなわちある程度の音量レベルとなっている可能性が高い。ボタン809が押されると、この特徴に基づいて、符号化のために選択されたPCMファイルの先頭部付近、あるいは最終部付近のデータの値を読み込み、それを分析することで該ファイルが他のファイルとの連続性を持つ可能性が高いか否かの判断がなされる。

【0101】分析の方法としては様々なものが考えられるが、最も単純な方法としては例えば、ファイルの先頭データと最終データを1ポイントずつ読み込み、それがゼロであるか否かの判断を行う。若し、先頭データがゼロ以外の値であれば、そのファイルは、他のファイルの最終部と連続性がある可能性があるものとする。若し、最終データがゼロ以外の値であれば、そのファイルは他のファイルの先頭部と連続性がある可能性があるものとする。これらの連続性がある可能性がある旨を使用者に警告するようにする。使用者は、警告に従い、該ファイルを試聴することで、連続性を確認することで、確実な判断を行うことができる。

【0102】また、例えば最終データがゼロ以外の値となるようなファイルが符号化処理の対象として選択されていた場合に、上述した警告を行わずに、符号化処理の際に、該ファイルの次に符号化処理されるように選択されたファイルと自動的に連続処理を行うようにしても良い。同様に、例えば先頭データがゼロ以外の値となるようなファイルが符号化処理の対象として選択されている場合に、上述した警告を行わずに、符号化処理の際に、該ファイルの前に符号化処理されるように選択されたファイルと自動的に連続処理を行うようにしても良い。

【0103】上述の説明では、先頭データと最終データを1ポイントずつで、しきい値をゼロとした場合の例について説明した。しかしながら、実際には音として聞こえても、アナログ的なノイズや、ディザ等の影響で、他のファイルと連続性がないのにも関わらず、先頭データまたは最終データがある程度の大きさの値をもったデータとなっている場合も少なくない。

【0104】この問題に対処するために、使用者が自由にしきい値の設定等を行えるようにする。また、しきい

値の設定は、レベルだけでなく、読み込みポイント数等でも行えるものとする。例えば、先頭データと最終データを1ポイントずつではなく10ポイントずつとし、その10ポイントの値の総和、平均値等をしきい値として用いる方法が可能である。あるいは、ファイルの最終値と他のファイルの先頭値との差分量を検索する方法等も可能である。このように、様々なしきい値設定を可能とすることで、より適切な判断が可能となる。

【0105】上述したボタン809がクリックされた場合の処理例の行程と、画面表示について図17および図18を参照して説明する。まず、ボタン809をクリックすると、ステップS51において検索条件の設定を行う。これはしきい値の設定を行うものである。例えば図18Aに示すような設定画面を通じて、先頭データと最終データについて読み込むデータのポイント数と、レベルを入力すると共に、しきい値の算出方法を入力するようにする。図18Aの例では、先頭データ、最終データ共に、10ポイントの平均値が5を上回るかどうかで判断する設定となっている。なお、算出方法については、平均値以外に総和、最大値等が選択可能とされている。

【0106】次に、ステップS52において、符号化処理選択ファイルを順に処理していくための変数の初期化が行われる。ここではこの変数を*i*として、ゼロを設定している。ステップS53では、符号化処理選択ファイル数と、変数*i*を比較することで、検索すべきファイルがあるか否かが決定される。ステップS53で、処理選択ファイル数を変数*i*が上回れば処理が終了となる（ステップS54）。

【0107】ステップS53において処理すべきファイルがあると決定されると、ステップS55では、該ファイルを読み込むためのオープン処理を行う。この後、ステップS56で、先頭部データを読み込む。ここで読み込むデータ数は、ステップS51で設定したポイント数となる。ステップS57では、ステップS51で設定した、しきい値の条件と読み込んだデータが比較される。

【0108】ステップS57で、読み込んだデータがしきい値を上回るようであれば、ステップS58の警告処理を行う。この警告処理では、例えば、図18Bに示すような警告メッセージを表示する。表示中のOKのボタンを使用者がクリックしてから次のステップに進む。ステップS59、S60、S61は、それぞれ最終部データについて、先頭データにおけるステップS56、S57、S58に対応した処理を行うものである。同様に、図18Cの警告メッセージの表示は、図18Bの先頭データに関する警告表示と対応するものである。

【0109】ステップS60またはS61までの処理を終えた後、ステップS62において、*i*番目のファイルをクローズ処理する。そして、ステップS63において、変数*i*のインクリメント処理を行い、ステップS53に戻る。この後に、警告のあったファイルについての



み試験を行うことで、使用者は符号化処理において連続性を考慮すべきか否かを容易に判断することが可能となる。また、ここでは符号化処理の前に、PCMファイルについて検索する例を述べたが、復号化の際にも同様に、符号化ファイルの正規化情報や、量子化状態などについてしきい値を持たせることで、同様の処理を行うことが可能である。

【0110】次に、検索結果をうけて、実際の符号化処理を行う。図11中のボタン807によって、図13に示した形で処理を行うか、図14に示した形で処理を行うかが選択される。なお、連続させるファイルの数が二つ以上の場合も同様である。連続処理させるファイルの選択については、図17で示した方法で割り出される。

【0111】図19は、連続させるファイルを実際に設定する方法の一例を示す。図19は、図11にてボタン807をクリックした場合に現れる操作表示画面であり、操作表示画面上で連続処理させるファイルが選択される。901で示す表示部には、連続処理を行うファイルを表示している。ここではdata2.pcmと、data3.pcmを連続処理する例が示されている。表示部901には、直接ファイル名を入力することが可能であるが、ボタン905を使って、いわゆるファイル構造をグラフィカルに検索し、ファイルを選択することも可能である。このとき、ファイルを選択した順序が、連続処理に反映されることとなるが、表示部901内で順序を変更することも可能である。

【0112】また、902を使用することで、複数のファイルの連続処理に対応することも可能である。表示部903は、表示部901と同様に、その他の組みで連続処理をさせるファイルについて設定するものである。この例ではdataA.pcm、dataB.pcm、dataC.pcmを連続処理させる設定が示されている。ここではdata2.pcmと、data3.pcmの連続処理を一組目、dataA.pcm、dataB.pcm、dataC.pcmを二組目としているが、とくにこの組の数値については、直接処理結果には関わらない。904については、一組目の902に相当するものである。また、ここでは二組を表示しているが、906を使用することで、このような組を、更に設定することも可能である。最後にOKボタン907をクリックすることで設定が完了する。

【0113】再び図11について説明する。806は、表示操作部804にて選択された高能率符号化データファイルを復号化する時に押されるボタンである。その処理方法、表示内容の対応等については、高能率符号化時のボタン805によるものと同様である。また復号化時においても、上述した高能率符号化の連続処理の場合と同様に、ボタン807を使用することで、連続復号化処理を設定することが可能である。復号化の連続処理の場合は、ある高能率符号化データファイルの最終フレーム

と、他の高能率符号化データファイルの先頭フレームを連続フレームとして復号化処理する形に設定を行うようにすればよい。808は、プログラムを終了させるためのボタンである。

【0114】上述した方法で、複数ファイルの符号化、復号化の際に、各ファイル独立に処理を行うか、または、異なるファイル間にまたがった連続性を考慮した処理を行うかを選択して、所望の形で処理ファイルを作成することが可能となる。また、連続性を考慮するか否かを、容易に判断することが可能となり、適応した処理をより迅速に行うことが可能となる。

【0115】

【発明の効果】上述したこの発明によるデジタル信号処理方法は、所望の複数ファイルの符号化処理を行う時に、異なるファイル間の始点、終点の連続性を考慮した符号化と、考慮しない符号化を選択することが可能である。また、この発明では、所望の複数ファイルの始点、終点のデータを分析し、分析結果をもとに連続性を考慮した符号化と、考慮しない符号化を選択することにより、より容易、かつ正確に、連続性を考慮した処理を行うか否かの判断ができるようになり、大幅な作業効率の向上が可能となる。

【図面の簡単な説明】

【図1】高能率符号化データの生成に係る構成の一例を示すブロック図である。

【図2】各帯域毎の直交変換ブロックサイズについて説明するための略線図である。

【図3】図1中の一部の構成について詳細に示すブロック図である。

【図4】臨界帯域、ブロックフローティング等を考慮して分割された帯域のスペクトルの一例を示す略線図である。

【図5】マスキングスペクトルの一例を示す略線図である。

【図6】最小可聴カーブ、マスキングスペクトルの合成について説明するための略線図である。

【図7】この発明の一実施形態における符号化データフォーマットの一例を示す略線図である。

【図8】図7中の1バイト目のデータの詳細を示した略線図である。

【図9】デジタル信号復号化処理に係る構成の一例を示すブロック図である。

【図10】符号化データ内の各フレームにおけるオーバーラップについて説明するための略線図である。

【図11】パソコン上で高能率符号化処理、および復号化処理を行うシステムの操作表示画面の一具体例を示す略線図である。

【図12】図11のシステムにより複数のファイルについて高能率符号化をおこなう処理を示す略線図である。

【図13】二つのファイルの連続性を考慮せずに高能率



符号化を行う場合のフレーム対応を示す略線図である。

【図 14】二つのファイルの連続性を考慮して高能率符号化を行う場合のフレーム対応を示す略線図である。

【図 15】二つのファイルの連続性を考慮せずに高能率符号化を行う場合の処理工程を示すフローチャートである。

【図 16】二つのファイルの連続性を考慮して高能率符号化を行う場合の処理工程を示すフローチャートである。

【図 17】先頭データと最終データを分析する処理工程を示すフローチャートである。

【図 18】先頭データと最終データを分析する処理工程において、検索しきい値の条件の入力画面、および警告メッセージの画面を示す略線図である。

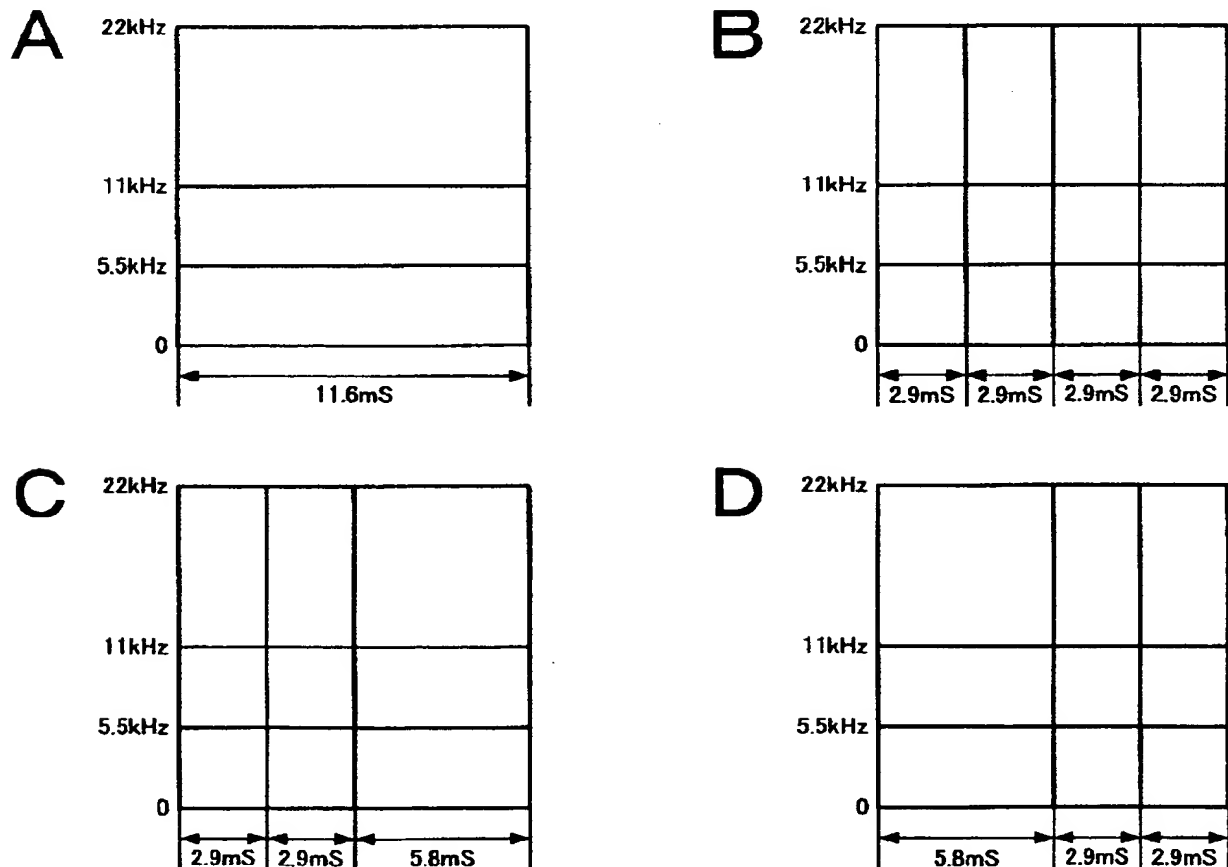
【図 19】連続性を考慮した処理を行うファイルの組合せを選択するための操作表示画面の一具体例を示す略線図である。

【符号の説明】

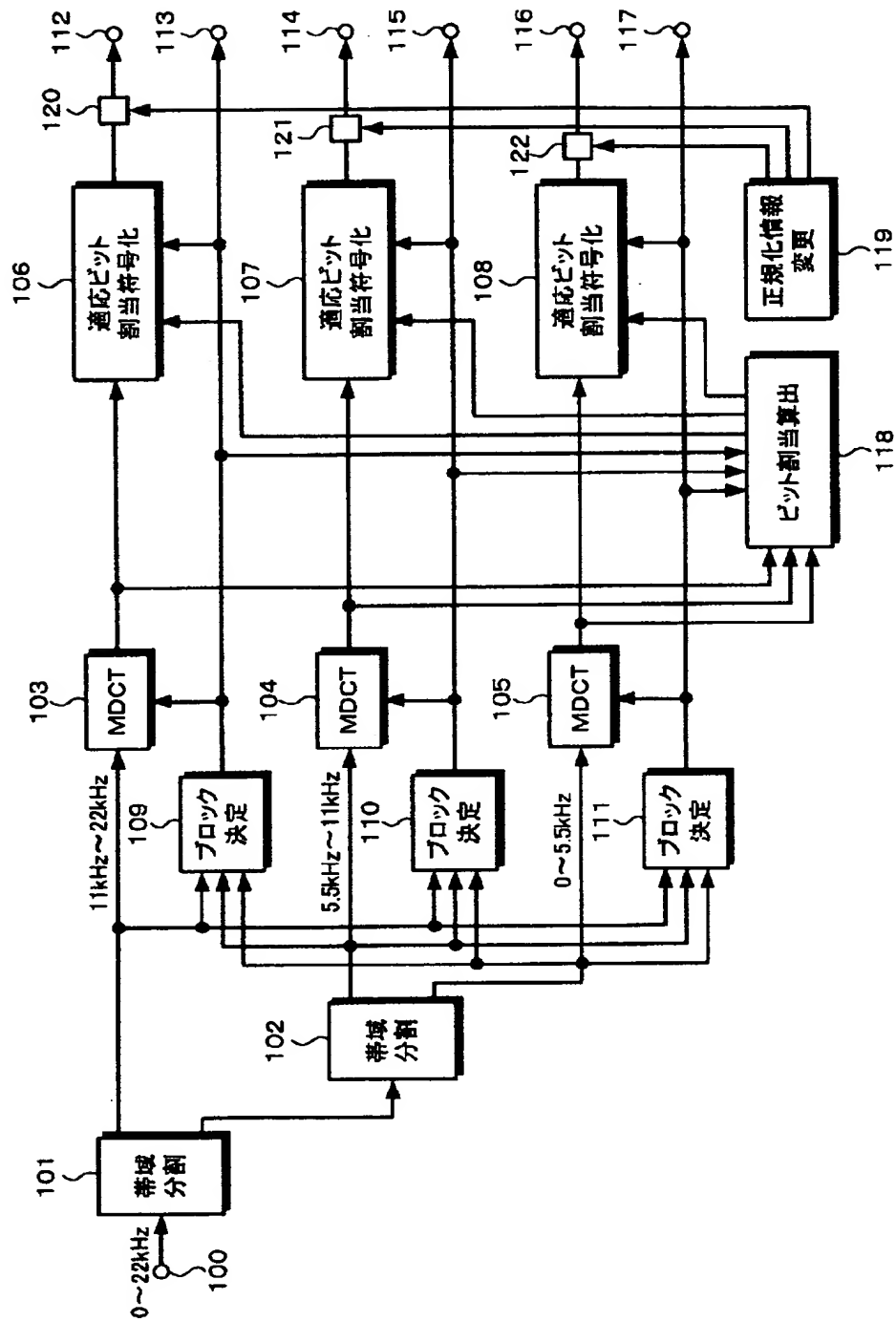
101、102・・・帯域分割フィルタ、103、10

4、105・・・直交変換回路 (MDCT)、109、110、111・・・ブロック決定回路、118・・・ビット割り当て算出回路、106、107、108・・・適応ビット割り当て符号化回路、119・・・正規化情報変更回路、120、121、122・・・加算器、302・・・帯域毎エネルギー算出器、303・・・畳込みフィルタ、304・・・加算器、305・・・関数発生器、306・・・割り算器、307・・・合成器、308・・・減算器、309・・・遅延回路、310・・・許容雑音補正器、701、702・・・帯域合成フィルタ (IQMF)、703、704、705・・・逆直交変換回路 (IMDCT)、706・・・適応ビット割当復号化回路、709・・・正規化情報変更回路、710・・・加算器、803・・・PCMデータファイルに関する表示操作部、804・・・符号化データファイルに関する表示操作部、807・・・複数のファイルの高能率符号化時の処理を選択するボタン、809・・・ファイルの連続性の解析を行うためのボタン

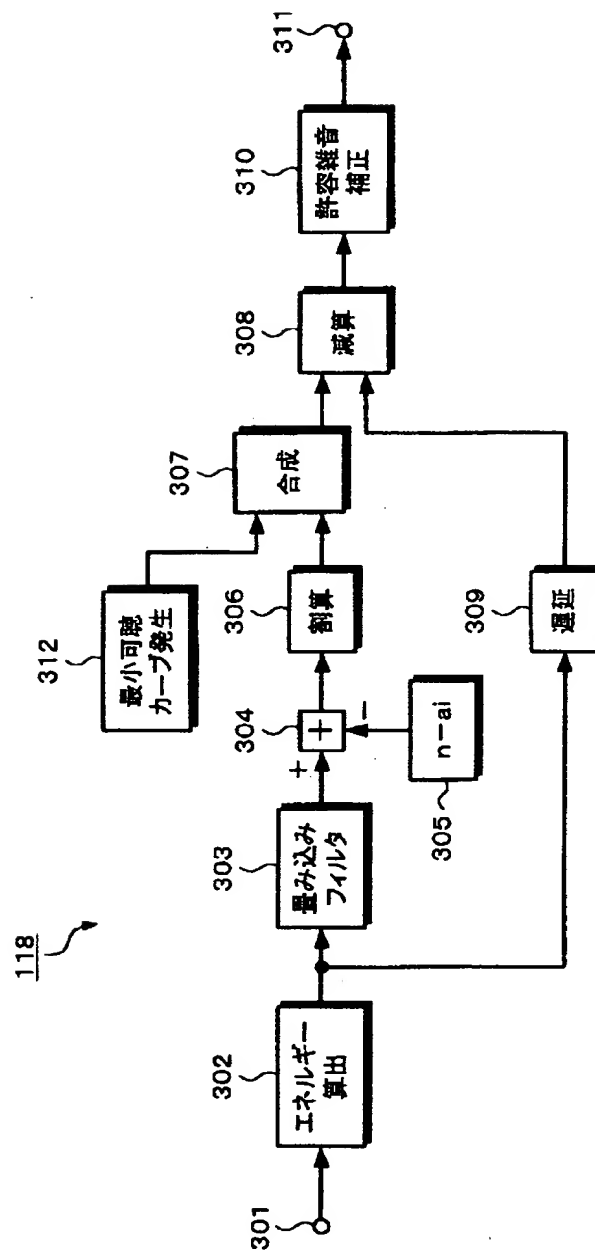
【図 2】



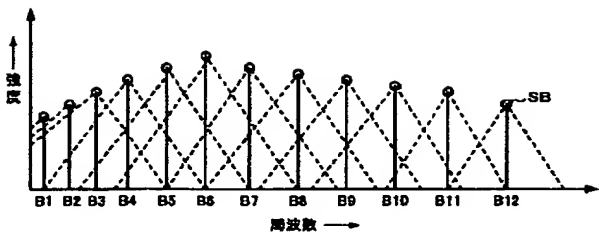
【図 1】



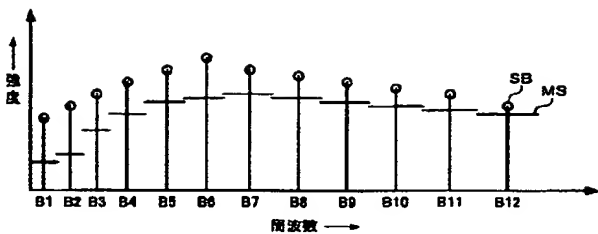
【図 3】



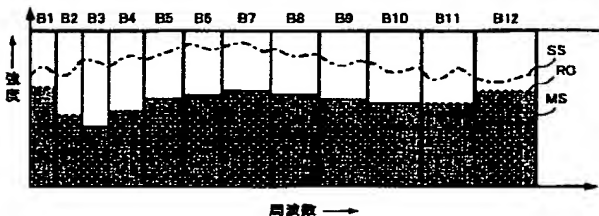
【図4】



【図5】



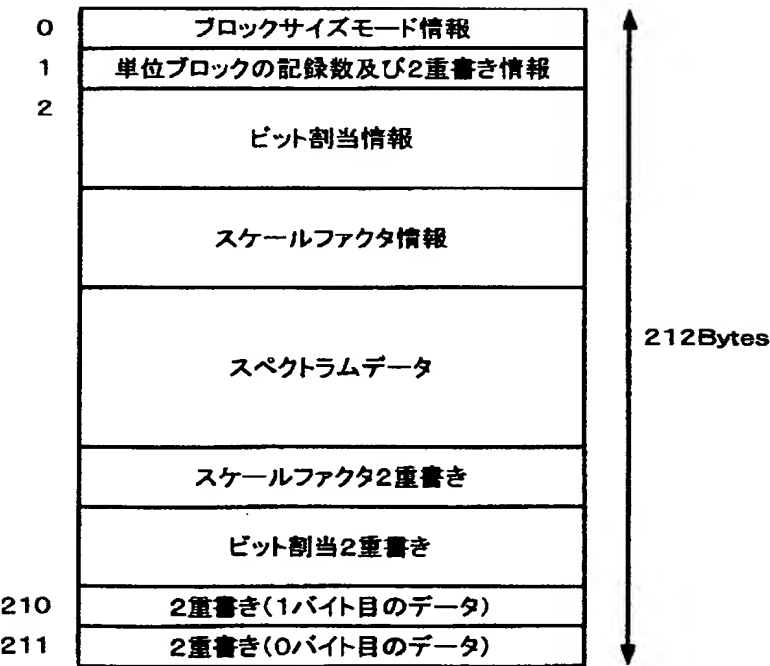
【図6】



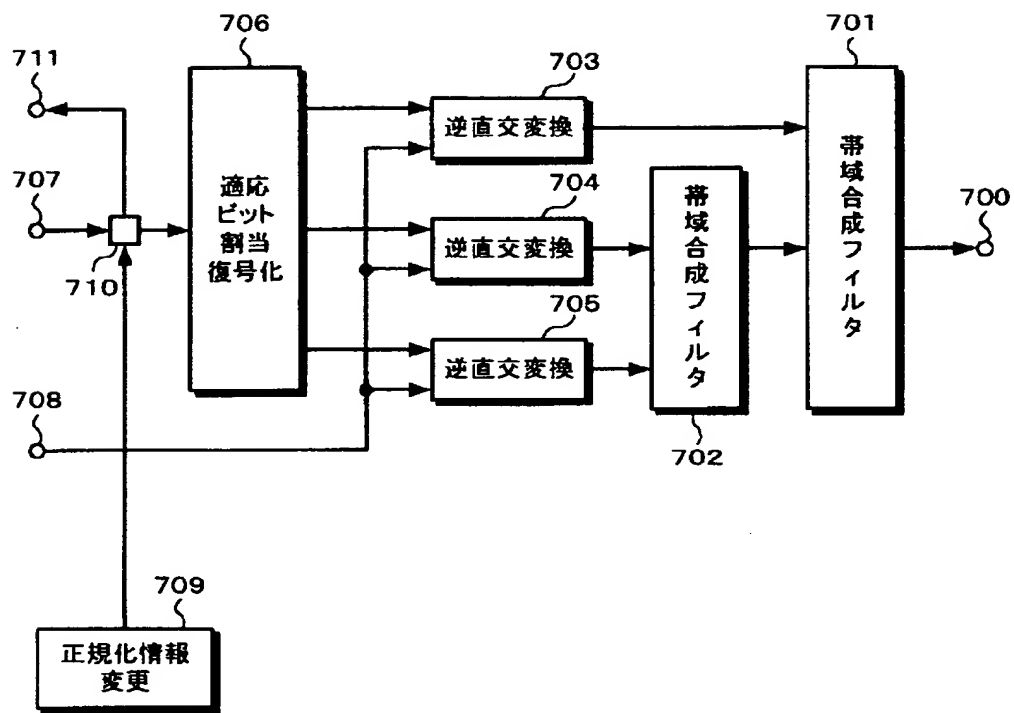
【図8】

記録する単位ブロックの個数		ビット割当情報の2重書き設定		スケールファクタ情報の2重書き設定	
コード	個数	コード	個数	コード	個数
000	20	00	0	000	0
001	28	01	28	001	8
010	32	10	44	010	12
011	36	11	52	011	16
100	40			100	24
101	44			101	36
110	48			110	44
111	52			111	52

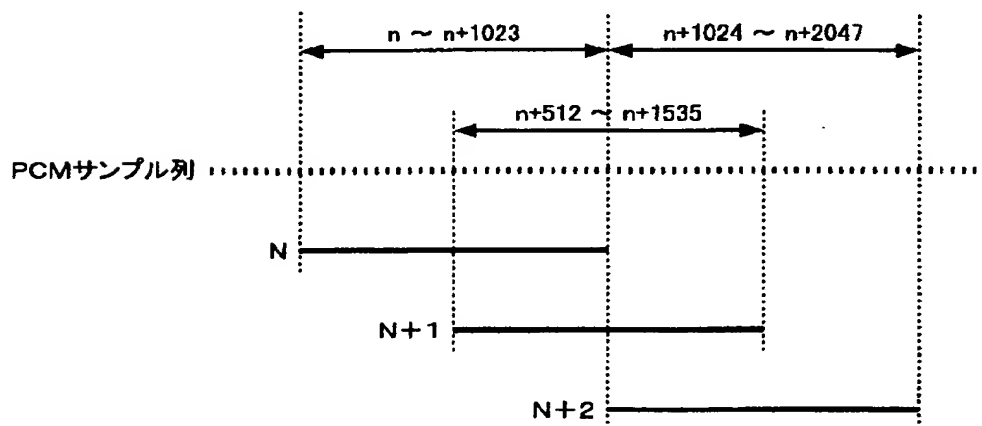
【図7】



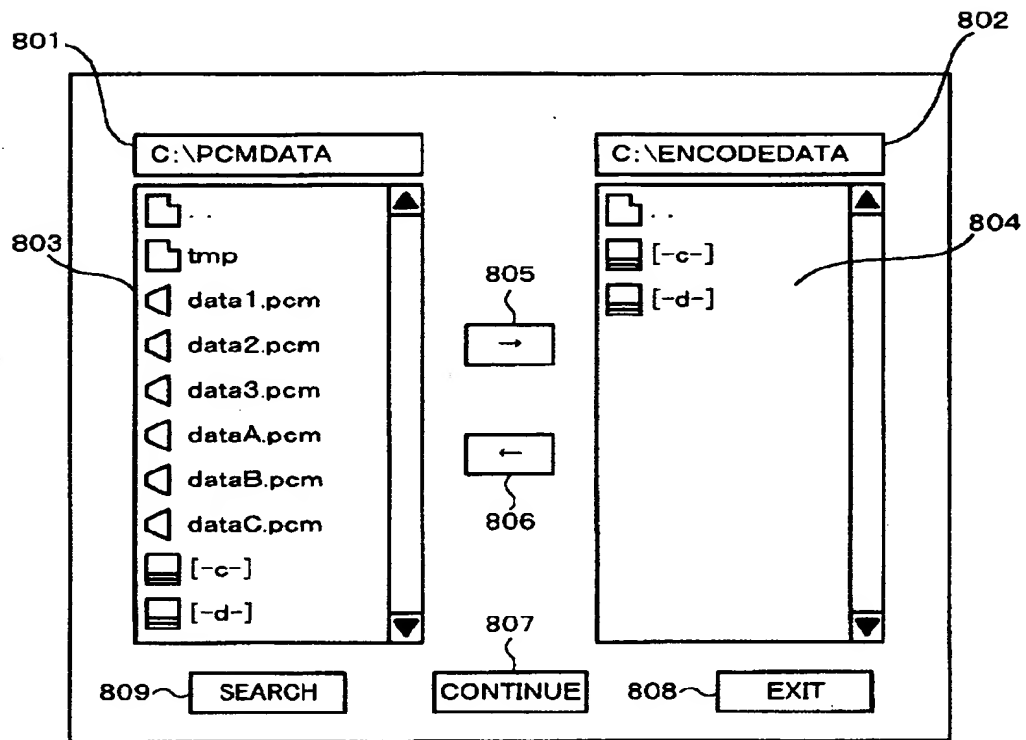
【図 9】



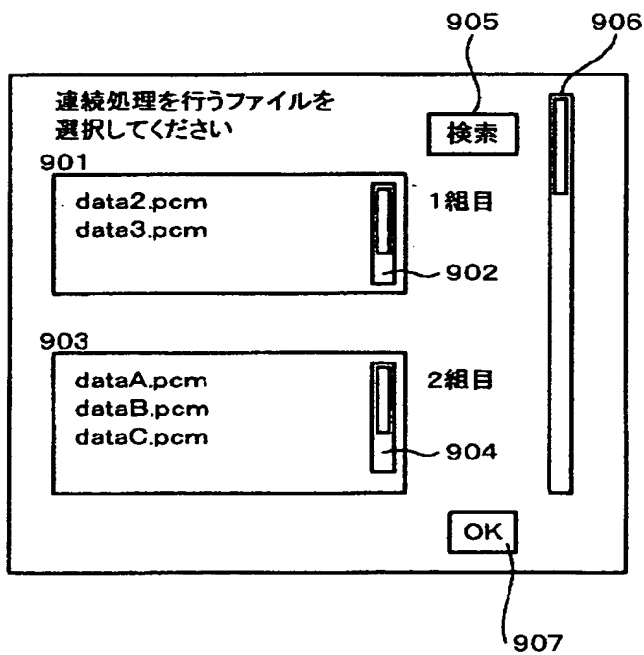
【図 10】



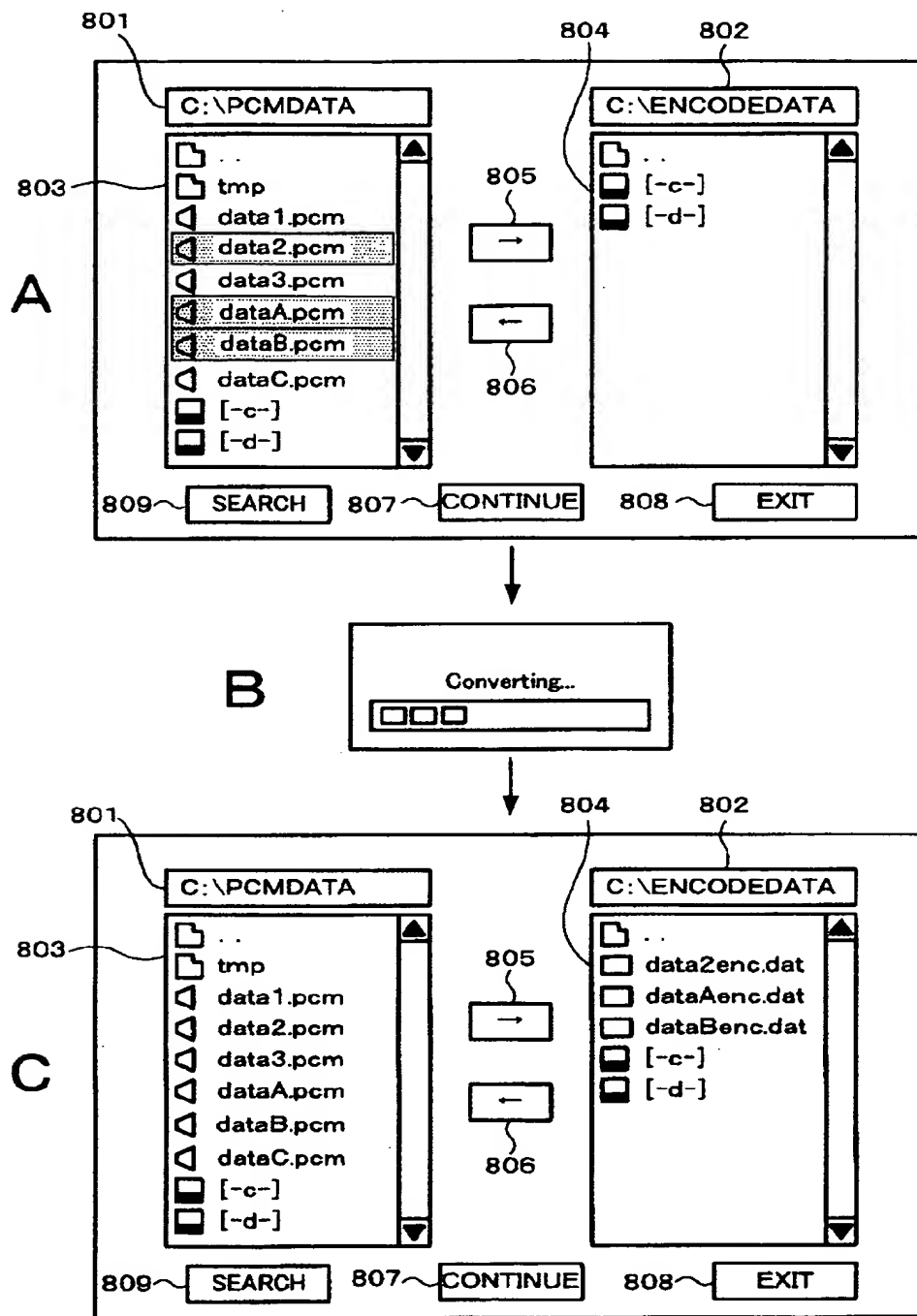
【図 11】



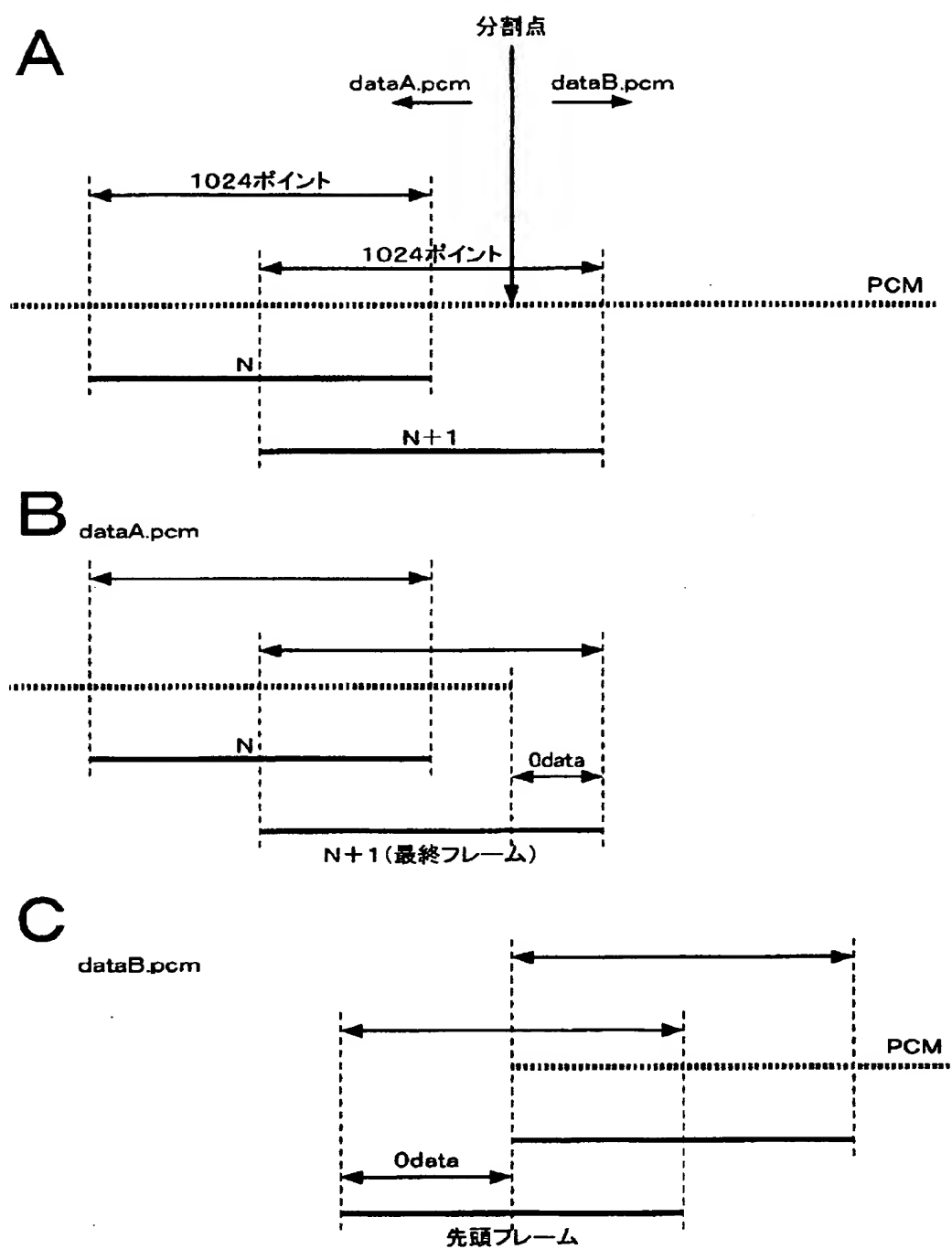
【図 19】



【図 12】

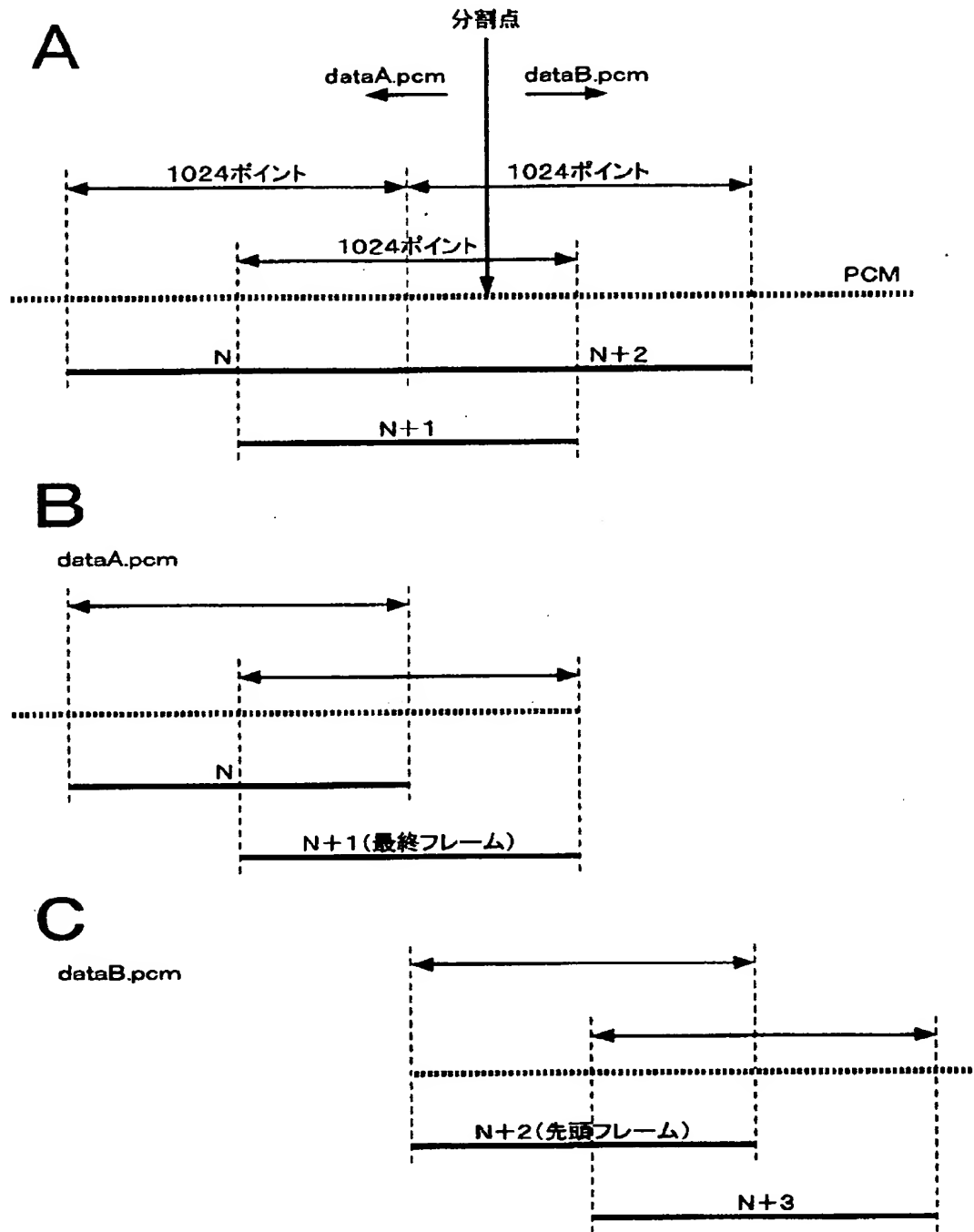


【図13】

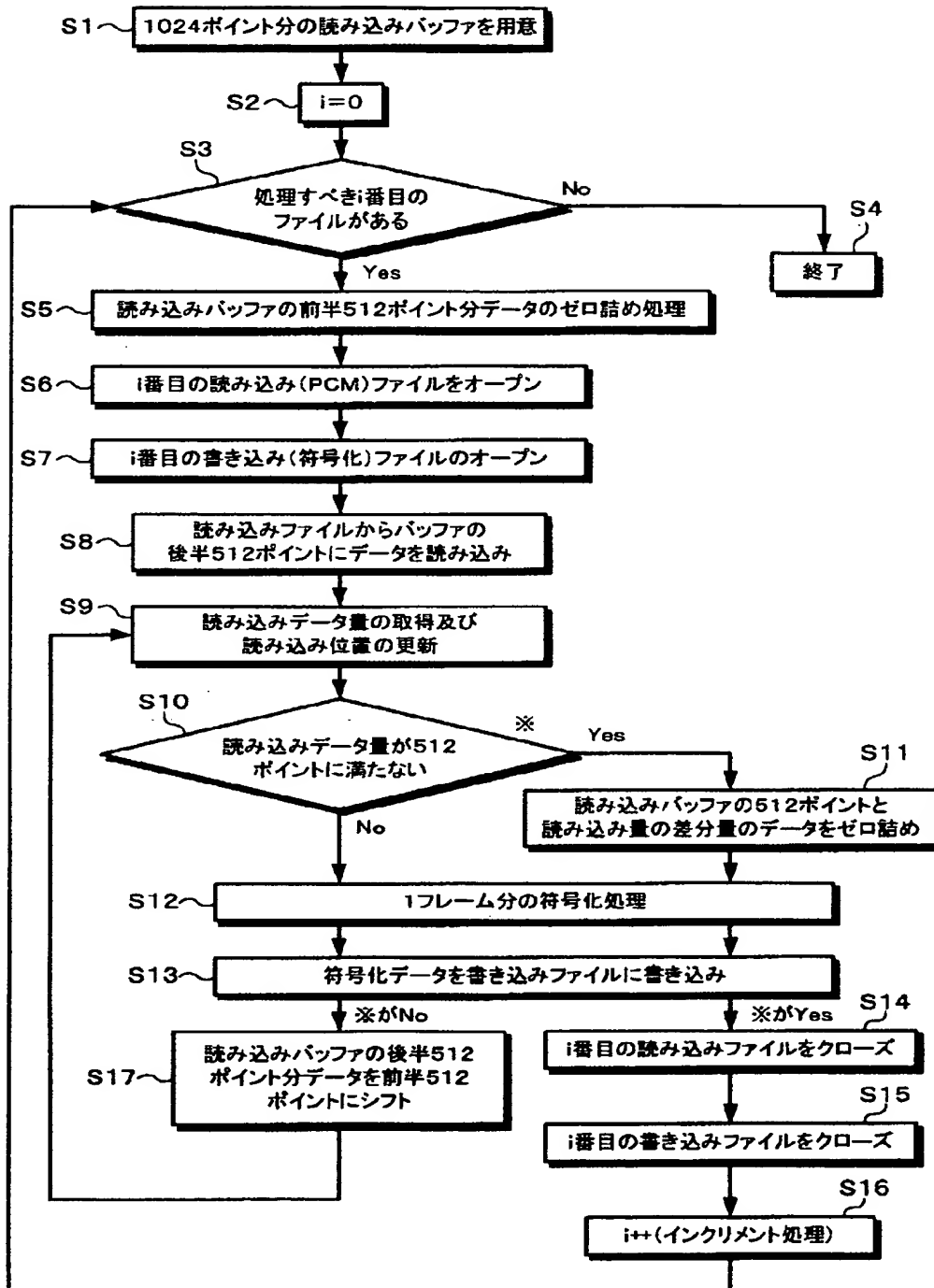




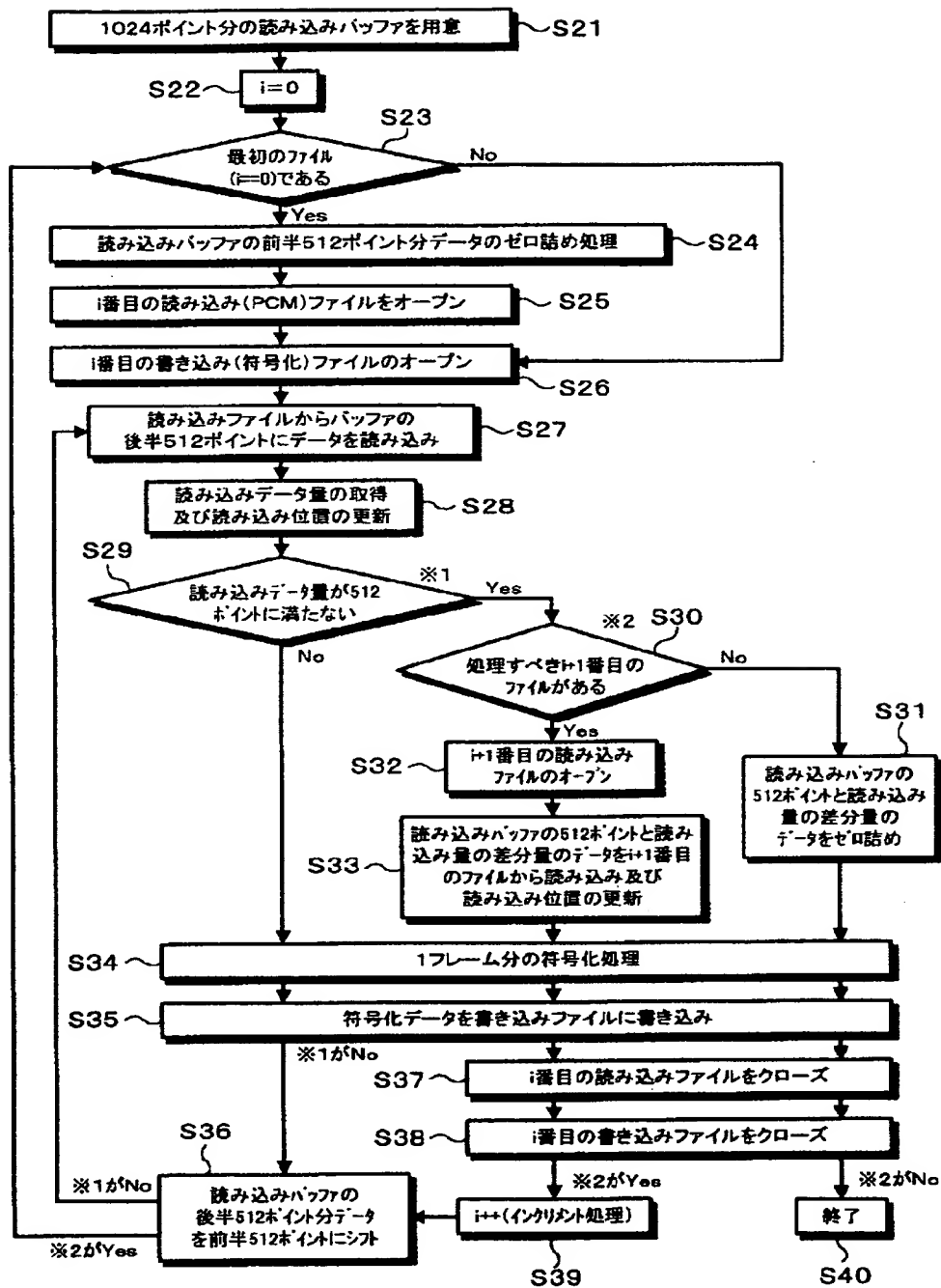
【図 14】



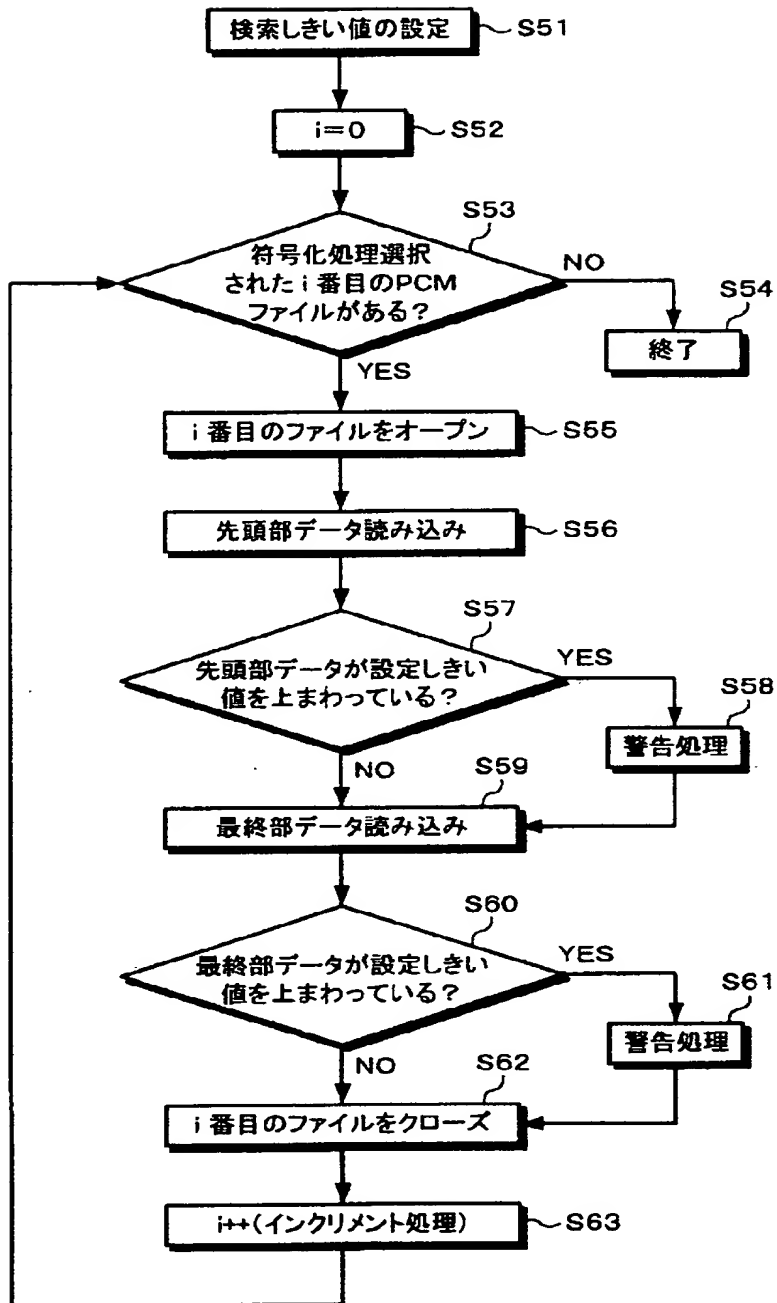
【図15】



【図 16】



【図17】



【図18】

A

検索したい値の条件を入力して下さい

	ポイント数	レベル	算出方法
先頭データ	<input type="text" value="10"/>	<input type="text" value="5"/>	<input type="text" value="平均"/>
最終データ	<input type="text" value="10"/>	<input type="text" value="5"/>	<input type="text" value="平均"/>

B

File\*\*\*. pcmは先頭データが  
無音ではありません。

C

File\*\*\*. pcmは最終データが  
無音ではありません。